



Research article

Multiobjective path optimization of an indoor AGV based on an improved ACO-DWA

Jinzhuang Xiao*, Xuele Yu, Keke Sun, Zhen Zhou and Gang Zhou

College of Electronic Information Engineering, Hebei University, Baoding 071000, China

* Correspondence: E-mail: robot@hbu.edu.cn.

Abstract: With their intelligence, flexibility, and other characteristics, automated guided vehicles (AGVs) have been popularized and promoted in traditional industrial markets and service industry markets. Compared with traditional transportation methods, AGVs can effectively reduce costs and improve the efficiency of problem solving in various application developments, but they also lead to serious path-planning problems. Especially in large-scale and complex map environments, it is difficult for a single algorithm to plan high-quality moving paths for AGVs, and the algorithm solution efficiency is constrained. This paper focuses on the indoor AGV path-planning problem in large-scale, complex environments and proposes an efficient path-planning algorithm (IACO-DWA) that incorporates the ant colony algorithm (ACO) and dynamic window approach (DWA) to achieve multiobjective path optimization. First, inspired by the biological population level, an improved ant colony algorithm (IACO) is proposed to plan a global path for AGVs that satisfies a shorter path and fewer turns. Then, local optimization is performed between adjacent key nodes by improving and extending the evaluation function of the traditional dynamic window method (IDWA), which further improves path security and smoothness. The results of simulation experiments with two maps of different scales show that the fusion algorithm shortens the path length by 9.9 and 14.1% and reduces the number of turns by 60.0 and 54.8%, respectively, based on ensuring the smoothness and safety of the global path. The advantages of this algorithm are verified. QBot2e is selected as the experimental platform to verify the practicability of the proposed algorithm in indoor AGV path planning.

Keywords: path planning; AGV; ant colony algorithm; dynamic window approach; multiobjective optimization

1. Introduction

With the increasing maturity of automation technology, an increasing number of automated guided vehicles (AGVs) has become the mainstream work method in the automatic handling process of factories. Its main function is to safely transport goods to the designated destination through the planned path in a complex working environment [1]. Compared with manual handling, AGVs can complete transportation tasks in harsh and complex environments. The emergence of the AGV automatic navigation system has changed the logistics structure of traditional manufacturing workshops, greatly reduced the production cost and improved the production efficiency of the enterprise.

The AGV navigation system can use AGVs to improve production and life efficiency in practical applications. However, it also produces a series of problems of environment perception, path planning and tracking control. Path planning is a key technology for mobile robot navigation and widely utilized in autonomous driving, logistics warehousing, AGV navigation and other fields [2]. Among them, the path-planning technology of AGVs is employed to identify a safe path in the free movement space of AGVs to satisfy one or more path performance indicators [3]. A reasonable and reliable driving path is crucial to ensure that AGVs can safely complete their handling tasks. To address the complexity of the environment and tasks in practical applications, increasingly many researchers are keen to solve the problem of AGV path planning, and path-planning technology has gradually become one of the research hotspots in AGV autonomous navigation systems.

With the continuous development of the AGV market and path-planning technology, increasingly many optimization algorithms are used to solve path-planning problems with different application backgrounds and task requirements. However, most of the previous works [4] considered the performance improvement of a single algorithm in small-scale, low-complexity maps and often ignored the limitations of a single algorithm in large-scale, complex environments, and there is a lack of practical application verification. In addition, people are often keen to consider a single path optimization objective of a single algorithm instead of the optimization of multiple path-planning indicators that conform to the actual AGV motion characteristics.

In this work, we mainly focus on indoor AGV path planning in large-scale and complex environments. In this paper, an efficient global path-planning algorithm is proposed to provide AGVs with high-quality paths that simultaneously satisfy multiple global path-planning performance evaluation indicators in large-scale and complex maps. The main contributions of this work are as follows:

1) Inspired by the ant colony algorithm, we design a state transition probability method to improve the search ability of ants, propose a pheromone update rule to increase the convergence ability of the algorithm and improve the quality of the optimal path solution, and consider the motion rules of AGVs in the real environment. We propose a key point path optimization strategy, which reduces the global path length and number of turns.

2) Based on the motion characteristics of AGVs, the evaluation function of the improved DWA algorithm is constructed, which considers the influence of obstacles on AGVs in a certain range, enhances the attraction of target points to AGVs, and improves the local optimization capability of the DWA algorithm.

3) Based on the improved ACO and DWA algorithms, an efficient fusion algorithm is proposed, which avoids the limitations of the two algorithms in solving path solutions and

optimizes various path optimization objectives.

The content of this paper is organized as follows: Section 2 presents the related work in this research area. Section 3 introduces the mathematical model of the traditional ant colony algorithm and construction basis of the improved ant colony algorithm in this paper. Section 4 introduces the working principle of the traditional DWA algorithm and construction basis of the improved DWA algorithm in this paper. Section 5 presents the detailed steps of the fusion algorithm in this paper. Section 6 verifies the performance of our improved algorithm from different aspects and compares it with existing methods. Section 7 summarizes the work of this paper and provides an outlook for the next steps.

2. Related work

The core of the AGV path-planning technology is the path-planning algorithm, which can be generally divided into global path planning and local path planning [5]. Global path-planning algorithms can be further divided into traditional algorithms [6–8] and intelligent optimization algorithms [9–11]. With the continuous development of artificial intelligence, many scholars have begun to use intelligent optimization algorithms [12]. As an efficient and stable intelligent algorithm, the ACO has the characteristics of distributed computing, ease of combination with other algorithms, and strong robustness [13]. However, ACO also shows obvious shortcomings in path-planning applications, such as slow convergence, ease of falling into a local optimal solution, and poor path quality [14]. For various path-planning problems, many scholars have given different solutions. Chen et al. [15] proposed a method based on adaptive clustering, which can provide a reasonable and efficient path solution for multiple heterogeneous UAVs in a large-scale coverage environment and greatly reduce the time required for search. Du et al. [16] proposed a linear formulation for path planning, which expands the diversity of algorithm search solutions and improves the maneuverability of the overall system. Considering various shortcomings of the ant colony algorithm in path planning, many scholars have attempted to improve it. Jiao et al. [17] designed a multi-state ant colony algorithm, which avoids the problem of local optimal solutions, but the convergence of the algorithm must be improved. Ling et al. [18] proposed an improved ant colony system algorithm for the UAV full-coverage path-planning problem, which can plan a reliable flight path for UAVs and improve the global search ability of the algorithm. Kumar et al. [19] applied the search and control of the optimal path of mobile robots by combining the ant colony algorithm and a heuristic technology. Various ant colony optimization algorithms have proposed substantial optimization schemes for path-planning problems [20]. The existing ant colony optimization algorithms usually do not consider the actual size of the AGV in solving path-planning problems, and the planned path security is relatively low. Moreover, the planned path does not satisfy the actual motion constraints of the AGV, and the solution efficiency of the algorithm on large-scale complex maps must be improved.

Local path planning is a commonly employed path-planning algorithm in dynamic environments, which includes the APF [21] and DWA [22]. The DWA is based on the motion characteristics of mobile robots and has good obstacle avoidance ability, and the planned path is relatively smooth. Jiang et al. [23] adaptively adjusted the evaluation function according to the state information of the robot, which alleviated the limitations of the DWA algorithm and shortened the path length. Ji et al. [24] proposed a new evaluation function that integrated the A* algorithm and DWA algorithm, which enables the robot to move smoothly, but the path global optimality is lacking. Lin et al. [25] added fuzzy control to the traditional DWA algorithm to evaluate the collision risk of obstacles and improved

the efficiency of the algorithm. Liu et al. [26] designed a feasibility evaluation function, improved the security of the path, and solved the problem of local optimal solutions. The DWA algorithm is a path-planning algorithm based on the motion characteristics of the robot. Although the planned path satisfies the safety and smoothness requirements of AGVs, in large-scale and complex environments, the path planned by DWA still has problems such as detours and unreachable target points due to the lack of global information.

As the task scenarios of path planning become more complex, the efficiency of solving path-planning algorithms in large-scale, complex maps remains challenging. As the path evaluation indicators become increasingly comprehensive, it is often difficult for an algorithm to plan a high-quality path that can simultaneously satisfy the requirements of multiple path optimization objectives. In this research, based on the existing problems of ACO and DWA algorithms in path-planning applications, a corresponding optimization scheme is designed to avoid its limitations, and a fusion algorithm for indoor AGV path planning is proposed. Compared with other studies, this study simultaneously optimizes multiple path objectives (path length, number of turns, smoothness and safety) in large-scale and complex maps.

3. Ant colony algorithm optimization

3.1. Traditional ant colony algorithm

In the traditional ant colony algorithm mathematical model, the state transition probability and pheromone update strategy have a crucial role in the solution efficiency of the algorithm [27].

3.1.1. State transition probability

At time t , ant k is moved from node i to node j by calculating the state transition probability $P_{ij}^k(t)$; then, the next node is selected according to the roulette method, which is defined as:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{j \in \text{allowed}_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}, & j \in \text{allowed}_k \\ 0, & \text{otherswise} \end{cases} \quad (1)$$

where $\eta_{ij} = \frac{1}{d_{ij}}$, $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, $\eta_{ij}(t)$ is the heuristic function and represents the Euler distance from the current node to the candidate node, $\tau_{ij}(t)$ is the pheromone concentration between two nodes, allowed_k is the set of next mobile nodes to be selected by the ants, and α and β are the pheromone importance factor and heuristic function importance factor, respectively.

3.1.2. Pheromone update strategy

The pheromone update strategy is a process in which the ant colony algorithm continuously realizes positive feedback. Through this method, the ant colony guides the descendants of the ants to continuously converge to obtain an optimal path. After all contemporary ants have reached the target, each ant will update the pheromone concentration according to Eqs (2)–(4).

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (3)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{if ant } k \text{ passes through path } (i, j) \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $\rho \in (0,1)$ is the pheromone volatilization coefficient, $\Delta\tau_{ij}(t)$ is the sum of pheromone increments for all ants, $\Delta\tau_{ij}^k(t)$ is the pheromone increment of the k^{th} ant on path (i, j) , Q is the pheromone intensity, and L_k is the total path length of the k^{th} ant after one iteration.

3.2. Improvement ant colony algorithm

3.2.1. Improvement heuristic function

The traditional ACO uses the reciprocal of the current node to the candidate nodes as the only heuristic factor, the difference between adjacent nodes is very small, and the guiding effect of the target point is very weak. Especially in large-scale environmental maps with many grids, the guiding effect of traditional heuristic functions is very poor, which is one of the main reasons for the poor quality of paths searched by ants.

The role of the heuristic function is to guide the ants to move toward high-quality nodes. An overly complex heuristic function increases the computational complexity of the algorithm, which affects the efficiency of the algorithm. Presently, most researchers use the inverse of the distance from the current node to the target point as the heuristic function. Therefore, this paper designs a new distance heuristic factor on this basis. Reducing the number of sharp corners and turns is simultaneously considered. In this paper, the angle ψ formed by the line between previous node f of the current node of the AGV and current node i and the line between current node i and next nodes j is selected as the direction heuristic factor to increase the heuristic information between its adjacent nodes. The improved heuristic function of the added direction factor is presented as follows:

$$\eta_{ij} = \frac{1}{ad_{je} + bC_{ij}} \quad (5)$$

where $d_{je} = \sqrt[3]{(x_e - x_j)^2 + (y_e - y_j)^2}$, which is the distance factor function, represents the distance from the candidate node to the target node. When d_{je} is smaller, the heuristic function value is larger to increase the heuristic information between its adjacent nodes. C_{ij} is the direction factor function; a and b are the weight coefficients to adjust the degree of influence of distance and direction, respectively; $a+b=1$.

A schematic of the direction heuristic factor is shown in Figure 1.

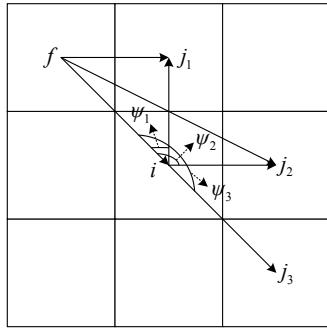


Figure 1. Schematic of the directional factor.

The direction factor function is presented as follows:

$$C_{ij} = \begin{cases} \frac{360^\circ}{\psi}, & \text{if } \psi \neq 180^\circ \\ 0.1, & \text{if } \psi = 180^\circ \end{cases} \quad (6)$$

When ψ is larger, the corner is smaller, i.e., the direction factor is smaller, so the probability of the node being selected is higher. When ψ is 180° , the three nodes are straight lines, i.e., there is no corner between the next node and the current node. At this time, the direction factor is extremely small, and the ants tend to choose this node.

3.2.2. Improvement state transition probabilities

The traditional ant colony algorithm uses the roulette method to select the next moving nodes of the ants. In this paper, an improved pseudorandom state transition probability strategy is applied to select the next node toward which the ants move. Under this strategy, the ants will preferentially choose the better path points to move in the early stage. During the continuous convergence process, the ants will tend to a reasonable path and avoid the probability of falling into the optimal solution. The improved state probability transition strategy is expressed as follows:

$$J = \begin{cases} \text{argmax} \{ [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta \}, & \text{if } q < q_0 \\ P_{ij}^k, & \text{otherwise} \end{cases} \quad (7)$$

where the function argmax is the maximum value of the product of $[\tau_{ij}(t)]^\alpha$ and $[\eta_{ij}(t)]^\beta$. When $q < q_0$, the state transition probability J is the maximum value of the product of the heuristic function and pheromone concentration; otherwise, it is the traditional roulette random selection mode.

In addition, this paper adopts the adaptive state transition parameters q_0 to adjust the probability of the next mobile node. The formula is presented as follows:

$$q_0 = \frac{NC}{N + NC} e^{(q_0-1)} \quad (8)$$

where q_0 represents the adaptive state transition parameters. When the algorithm starts running, $q < q_0$, the ants tend to the path node with the maximum product of the two as the next mobile node. This is a deterministic selection mode, which increases the convergence. In contrast, with an increase

in number of iterations, q_0 continues decreasing, and the path selection is more inclined to the traditional roulette random selection mode, which increases the possibility of ants choosing different paths and further improves the ability of the algorithm to explore new paths.

Reference [28] obtains the optimal parameter configuration through the control variable method. The values of α and β influence the solution result. Under the constant action of the positive feedback of the algorithm, the ants can easily converge to the local optimal solution. This paper adopts an improved adaptive greedy policy parameter adjustment method [29]. The ability of pheromone concentration and heuristic information to influence the path search is continuously adjusted when the number of iterations increases, and parameters α and β will be nonlinearly accelerated during each iteration by Eq (8). This approach ensures that the path is fully excavated during the process of ant colony search, increases the global exploration performance of the algorithm, and avoids the problem of local optimal solutions.

$$\begin{cases} \beta = \beta_{\max} - (\beta_{\max} - \beta_{\min}) \left(\frac{N}{NC} \right) \\ \alpha = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \left(\frac{N}{NC} \right) \end{cases} \quad (9)$$

where the maximum value and minimum value of the heuristic information impact factor are β_{\max} and β_{\min} , respectively; the maximum value and minimum value of the pheromone influence factors are α_{\max} and α_{\min} , respectively; NC is the total number of iterations; N is the current number of iterations.

3.2.3. Improvement pheromone update strategy

In the traditional ACO, after each iteration, each ant updates the pheromone according to its path length. When the path length difference is small, the pheromone difference becomes very small, and it is difficult for the descendant ants to search for a new better solution, which affects the performance of the algorithm optimization. Therefore, this paper proposes a dynamic, grading-differentiated, pheromone update strategy to increase the convergence speed. Among them, the dynamic grading strategy [30] and differentiated pheromone update strategy are two main parts of the method.

1) Dynamic grading strategy

In biological populations, there is a strict hierarchical system in which high-level individuals have higher priority and can guide low-level individuals. Inspired by the biopopulation system, this paper calculates the fitness of each ant according to Eqs (10) and (11) to dynamically classify ants. Three different levels of ants coordinate search for the optimal path.

$$fit_i = \frac{L_b}{L_k} \quad (10)$$

$$Rank = \begin{cases} \text{first-level ants,} & \text{if } fit=1 \\ \text{second-level ants,} & \text{if } R < fit < 1 \\ \text{third-level ants,} & \text{otherwise} \end{cases} \quad (11)$$

where L_b is the contemporary optical path length, L_k is the path length of the current k^{th} ant, fit

is the fitness of each ant, and $R \in (0,1)$ is the threshold of the first-level ants and second-level ants.

2) Differentiated pheromone update strategy

During each iteration, the quality of the paths found by each ant is different and has certain differences. If each ant updates the pheromone of the paths found by itself, then some poorer paths may cause interference to the search of the descendant ants. Meanwhile, if only the pheromone on the optimal path is updated, the exploration of some optimal paths may be lost. According to the difference in path lengths searched by different ants in each iteration process, the pheromone increments on different paths are adjusted. The update rules are shown in Eqs (12)–(14):

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (12)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (13)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q \cdot fit_i}{L_b} + \frac{Q \cdot fit_i}{L_a} - \frac{Q \cdot fit_i}{L_w}, & \text{if first-level ants} \\ \frac{2 \cdot Q \cdot fit_i}{L_k} - \frac{Q \cdot fit_i}{L_w}, & \text{if second-level ants} \end{cases} \quad (14)$$

where L_b is the optimal path length for one iteration, L_w is the worst path length, and L_a is the mean of the path lengths of each ant.

In addition, to further increase the convergence speed and make the selection of pheromones for offspring ants more directional, the idea of “the law of the jungle” in the wolf pack algorithm is applied to encourage the optimal path in each iteration process, and the amount of pheromones on its path is increased. In contrast, the worst path is penalized, and the amount of pheromones on its path is reduced. Next, the final global pheromone update rule is expressed as follows:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}^+ - \Delta\tau_{ij}^- \quad (15)$$

$$\Delta\tau_{ij}^+ = \begin{cases} \frac{Q}{L_b}, & \text{if } (i, j) \in L_b \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

$$\Delta\tau_{ij}^- = \begin{cases} \frac{Q}{L_w}, & \text{if } (i, j) \in L_w \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

The pheromone on the paths of ants of different levels is dynamically updated by Eq (15), and the differentiated path information of the ant colony in the iterative process is fully utilized to increase the guiding effect of the better ants in the ant colony on the offspring ants and to reduce the poor interference of ants on the path search.

3.2.4. Key node path optimization strategy

Restricted by the grid map, the AGV can only move in eight different directions. In Figure 2, the path planned by the ACO is only the optimal path based on the grid map movement rules, not the actual

shortest path, and the planned path does not conform to the actual movement rules of the AGV.

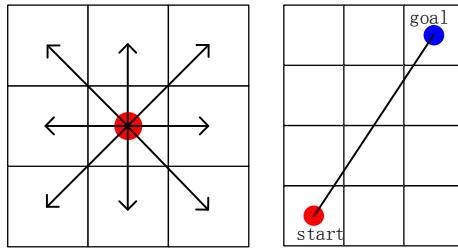


Figure 2. Schematic of the AGV moving direction.

As mentioned above, this paper proposes a key node path optimization strategy. The detailed process is as follows:

Step 1: Enter the path points obtained by the ant colony algorithm $U = \{P_i, 1 \leq i \leq n\}$;

Step 2: Add the path starting point P_1 as the first starting node P_s to the key node set k_path ;

Step 3: Sequentially connect the path points $P_i (3 \leq i \leq n)$ from the starting node and determine whether the straight line $P_s P_i$ has passed through the obstacle. If it passes through the obstacle, add $P_i - 1$ as the key node set k_path . Then, P_i is used as the new starting node to continue connecting the waypoints after P_i . If the line connected by $P_s P_i$ does not pass through the obstacle, continue to connect the path points in the path set U backward until the end point P_e is obtained and add it to k_path as the last key node, where $k_path = \{P_s, P_i - 1, P_3, \dots, P_{m+k}, P_n\}$;

Step 4: The key nodes in the set k_path are connected in turn, and the algorithm ends.

4. Dynamic window approach optimization

4.1. Traditional dynamic window approach

The DWA algorithm converts the position control of the AGV into speed control, samples multiple sets of speeds in the speed space at time t , simulates and predicts the trajectory of the AGV at time Δt at these speeds, selects the optimal trajectory at time $t + \Delta t$ according to the evaluation function, and controls the robot motion with the corresponding speed command [31].

4.1.1. AGV motion model

First, the kinematic model of the AGV must be analyzed. Assuming that the AGV moves along a straight line at time Δt , the trajectory of the AGV at time Δt can be presented as:

$$\begin{cases} x_{t+1} = x_t + v\Delta t \cos(\theta_t) \\ y_{t+1} = y_t + v\Delta t \sin(\theta_t) \\ \theta_{t+1} = \theta_t + \omega \Delta t \end{cases} \quad (18)$$

where x_t , y_t and θ_t are the coordinates of the AGV in the global coordinate system at time t .

4.1.2. Speed sampling

AGV has multiple sets of speeds (v, ω) in the speed space, but the final speed will be constrained to a certain speed window.

1) Maximum and minimum speed limits

According to the moving speed of the robot, let \mathbf{V}_m be denoted as the maximum and minimum speed space; its form is presented as follows:

$$\mathbf{V}_m = \{(v, \omega) \mid v_{\min} \leq v \leq v_{\max}, \omega_{\min} \leq \omega \leq \omega_{\max}\} \quad (19)$$

where v is the linear velocity, ω is the angular velocity, v_{\min} and ω_{\min} are the minimum linear velocity and minimum angular velocity, respectively, and v_{\max} and ω_{\max} are the maximum linear velocity and maximum angular velocity, respectively.

2) Acceleration constraints

Limited by the torque of the motor, the AGV is constrained by the minimum acceleration. Let \mathbf{V}_d be denoted as the speed space that can be achieved at the next moment; its form is expressed as follows:

$$\mathbf{V}_d = \{(v, \omega) \mid v_c - \dot{v}_b \Delta t \leq v \leq v_c + \dot{v}_a \Delta t, \omega_c - \dot{\omega}_b \Delta t \leq \omega \leq \omega_c + \dot{\omega}_a \Delta t\} \quad (20)$$

where v_c and ω_c are the current speed, \dot{v}_a and $\dot{\omega}_a$ are the maximum acceleration, and \dot{v}_b and $\dot{\omega}_b$ are the maximum deceleration.

3) Safety constraint

AGVs should brake when approaching an obstacle to avoid the risk of a collision. Let \mathbf{V}_a be denoted as the safe speed space of the AGV; its form is expressed as follows:

$$\mathbf{V}_a = \{(v, \omega) \mid v \leq \sqrt{(2 \text{dist}(v, \omega) \dot{v}_b)}, \omega \leq \sqrt{(2 \text{dist}(v, \omega) \dot{\omega}_b)}\} \quad (21)$$

where $\text{dist}(v, \omega)$ is the closest distance between the simulated trajectory and the obstacle.

The final speed is constrained in three sets; i.e., the dynamic window \mathbf{V}_z can be expressed as follows:

$$\mathbf{V}_z = \mathbf{V}_m \cap \mathbf{V}_d \cap \mathbf{V}_a \quad (22)$$

4.1.3. Evaluation function

The dynamic window approach evaluates the predicted trajectory through the evaluation function and outputs the optimal speed that satisfies the constraints. The traditional evaluation function is expressed as follows:

$$G(v, \omega) = \delta[\sigma \cdot \text{heading}(v, \omega) + \tau \cdot \text{vel}(v, \omega) + \gamma \cdot \text{odist}(v, \omega)] \quad (23)$$

where $\text{heading}(v, \omega) = |180^\circ - \theta|$ is the direction evaluation function; θ is the direction angle between the AGV movement direction and the target point; $\text{vel}(v, \omega)$ is the speed evaluation function, which generally refers to the linear speed of the AGV; $\text{odist}(v, \omega)$ is the obstacle distance evaluation function, which represents the shortest distance from the end of the predicted trajectory to the obstacle;

σ , τ , and γ are the weight coefficients of each evaluation function; δ is the evaluation function that indicates that the evaluation function is normalized.

In summary, the evaluation function is an important factor that affects the quality of the trajectory, which aims to make the AGV avoid obstacles as much as possible and reach the target at the fastest speed.

4.2. Improvement dynamic window approach

The dynamic window approach fully considers the AGV motion characteristics, and the planned path is smoother and safer. However, in large-scale complex environments, due to the unreasonable mechanism of the traditional DWA evaluation function, there are often problems of falling into local optima and unreachable goals. In this paper, the local optimization ability of the DWA algorithm is improved by improving and extending the evaluation function of the traditional DWA. The specific design is presented as follows:

1) Improvement obstacle evaluation function

In the traditional DWA algorithm, to ensure the safety of the path, the weight of the obstacle distance evaluation function generally accounts for a large proportion, i.e., the algorithm always preferentially selects the path point far from the obstacle, as shown in Figure 3.

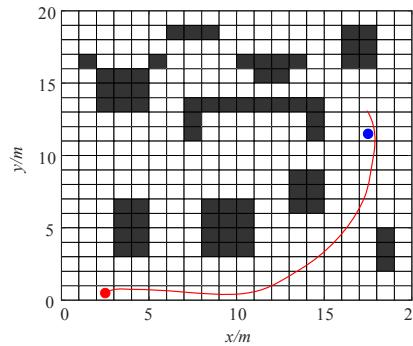


Figure 3. Path planned by the traditional DWA algorithm.

When the AGV is in a wide environment or there are no obstacles between it and the target point, the trajectory far from the obstacle will receive a higher score, which is one of the main reasons for the inaccessibility of traditional DWA detours and target points. Considering this situation, this paper improves the traditional obstacle distance evaluation function. In addition, to ensure the safety of the planned path, a safety distance R is introduced. Only the obstacles within a certain distance are considered, and the obstacle distance evaluation is limited to a maximum value to prevent a predicted trajectory from obtaining a higher score due to being far away from the obstacle. The improved obstacle distance evaluation function is as follows:

$$\begin{cases} dist(i,o) = \min \{ dist(p,o) \} \\ odis(v,\omega) = \begin{cases} 0 & \text{if } dist(i,o) \leq R \\ dist(i,o), & \text{if } R < dist(i,o) > 2R \\ 2R, & \text{if } dist(i,o) > 2R \end{cases} \end{cases} \quad (24)$$

where $o \in \{ob_1, ob_2, \dots, ob_m\}$ is the position of the obstacle, p is the position of the end of the predicted trajectory, $dist(i, o)$ is the predicted distance between the end of the trajectory and the nearest obstacle, $R = \frac{\sqrt{2}}{4}L$ is a safe distance, and $L = lm$ is the side length of the map grid. From Eq (24), the predicted trajectory will be constrained within a certain safety distance from the obstacle.

2) Adding the key node distance evaluation function

In addition, if the target point in the environment is near the obstacle, the existing evaluation function will make the AGV wander near the target point and fall into a local optimum, as shown in Figure 4.

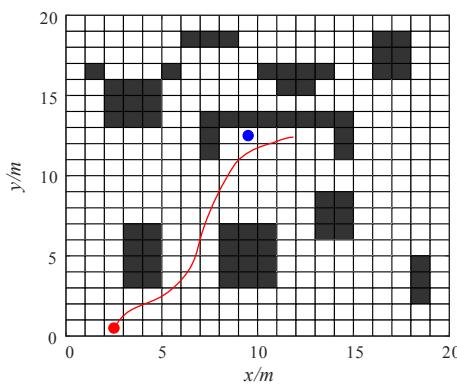


Figure 4. Plan path when $sgdist(v, \omega)$ is not added.

Considering this special situation, this paper introduces a target distance function [32], based on which the key node distance function is designed to increase the navigation ability of the target point. The formula is as follows:

$$sgdist(v, \omega) = \begin{cases} dist(S, E) - dist(p, kpi), & \text{if } n \geq 3 \\ dist(S, E) & \text{otherwise} \end{cases} \quad (25)$$

where $dist(S, E)$ is the distance from the starting point to the target point, $dist(p, kpi)$ is the distance between the end of the predicted trajectory and the key node, and n is the number of key nodes. As shown in Figure 5. When the target point or key node is too close to the obstacle, $\theta_1 < \theta_2$ and $d_{o1} < d_{o2}$. The traditional evaluation function will cause the two paths to obtain similar scores. At this time, the AGV may choose path B, which makes the target unreachable. After adding the $sgdist(v, \omega)$ function, $del < de2$, and Path A will obtain a higher score to guide the robot toward the target point. Equation (25) shows that on any scale map, the path closer to the key node will obtain a higher evaluation score, which increases the ability of the robot to navigate to the target point.

The function $sgdist(v, \omega)$ focuses on the guidance of the AGV that approaches the subtarget point. When the AGV is far from the target point, $heading(v, \omega)$ plays a leading role in guiding, and the two functions will be in different movement stages to guide the AGV to reach the target.

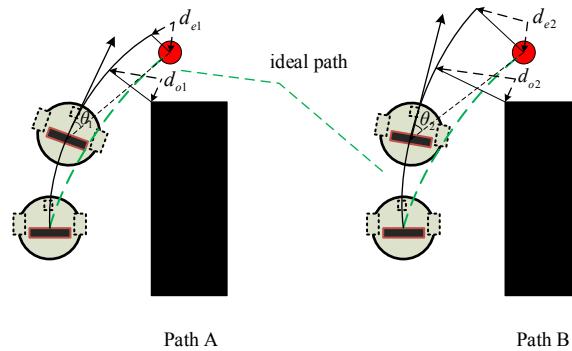


Figure 5. Functional analysis for the function $sgdist(v, \omega)$.

As mentioned above, the final evaluation function of the improved DWA algorithm in this paper is expressed as follows:

$$IG(v, w) = \delta[\sigma \times Iheading(v, \omega) + \tau \cdot vel(v, \omega) + \gamma \cdot Iodist(v, \omega) + \varepsilon \cdot sgdist(v, \omega)] \quad (26)$$

where $Iheading(v, \omega) = |180^\circ - \theta|$; θ is the direction angle between the AGV and the next key node, not the direction angle with the final target point; the initial direction of the AGV is set as the direction angle between the AGV and the first key node. $Iodist(v, \omega)$ is an improved obstacle distance evaluation function, and $sgdist(v, \omega)$ is a key point distance evaluation function.

5. Fusion algorithm

Aiming at the existing problems in the traditional ACO and DWA in large-scale and complex environments and comprehensively considering the advantages and disadvantages of the two algorithms, a fusion algorithm is proposed to achieve AGV multiobjective path optimization. First, by improving the heuristic function of the ant colony algorithm, the number of turns of the path is reduced; an improved adaptive state transition strategy is adopted to prevent the algorithm from falling into the local optimal solution; a dynamic, grading-differentiated, pheromone update strategy is proposed to increase the convergence speed. Then, an improved key node path optimization strategy is adopted to simplify redundant nodes to obtain a global path that can better satisfy the actual motion rules of the AGV. Finally, to improve the problems of detours and unattainable goals in the planned path, by improving and extending the evaluation function of the traditional dynamic window method, the adjacent key nodes are locally optimized until the AGV reaches the destination. The safety and smoothness of the path are further improved. In summary, a high-quality path can be planned for the AGV that satisfies the short path, fewer turns, and high safety and smoothness.

The specific steps of the IACO-DWA are described as follows:

Step 1: Build a grid map, set the start and target points for the AGV, and initialize the algorithm parameters;

Step 2: Use the IACO to plan a global optimal path including key nodes;

Step 3: Select key nodes as local subgoals of the improved DWA algorithm;

Step 4: Confirm the current starting point and subtarget position of the AGV and use the improved DWA for speed sampling and trajectory prediction;

Step 5: Evaluate the predicted trajectory based on Eq (26) and output the optimal trajectory that satisfies the constraints;

Step 6: Determine whether the local subtarget point is reached; if it is, perform the following steps. Otherwise, return to Step 4;

Step 7: Determine whether the target point is reached; if it is, output the path. Otherwise, return to Step 4.

6. Simulation and experimental verification

The experimental running environment is described as follows: Windows 10 (64-bit); Intel® Core™ i7-7500U; memory: 8 GB; MatLab R2019a.

This paper conducted experimental verification from the following aspects: 1) The gradient comparison experiment was performed on a 20×20 scale grid map (environment 1) to verify the effectiveness of each step of IACO. The specific process of the gradient comparison experiment is presented as follows. Experiment 1: The improved adaptive pseudorandom state transition probability was added based on the traditional ACO. Experiment 2: The improved heuristic function of this paper was added based on Experiment 1. Experiment 3: Based on Experiment 2, the dynamic grading-differentiated pheromone update strategy proposed in this paper was added. Based on Experiment 3, the key node optimization strategy was adopted to obtain the IACO of this paper. Then, the IACO in this paper was compared with the traditional ant colony algorithm, the ant colony algorithm in reference [33] and other swarm intelligence algorithms [34]. 2) In a 20×20 scale raster map (environment 2), the IDAW was used for path planning to verify its effectiveness. 3) In two grid maps of different scales, the advantages of IACO-DWA were verified by comparing the traditional algorithm and the algorithms in references [33] and [35]. 4) The QBot2e mobile robot was utilized as the test object to verify the practicability of the IACO-DWA in an actual environment. These experiments fully verify the performance of the single algorithm and the overall ablation results of the fusion algorithm in the experiment.

6.1. Gradient comparison experiment simulation analysis

The parameters of the traditional ACO are as follows: m is 50, NC is 100, α is 1, β is 7, ρ is 0.3, and Q is 1. In the following experiments, the algorithms in reference [33] and reference [34] have consistent parameters with the original reference. The parameters of the IACO are as follows: m is 50, and NC is 100. According to reference [28], the optimal parameter range is obtained by the control variable method; β_{\max} and β_{\min} are 8.5 and 2.5, respectively; α_{\max} and α_{\min} are 1.5 and 0.5, respectively; ρ is 0.3; Q is 1; $q_0 \in (0,1)$; q_0 is 0.9; a and b are the distance weight and direction factor weight, respectively. Since the shortest distance is given priority in ant pathfinding, a and b are 0.7 and 0.3, respectively. Considering that the map environment in this paper is relatively complex, to more rapidly obtain a better solution of the path, R is set to 0.95. To reduce the accidental nature of the experiment, each experiment is performed 20 times to draw a comparison table of algorithm performance for a comprehensive evaluation.

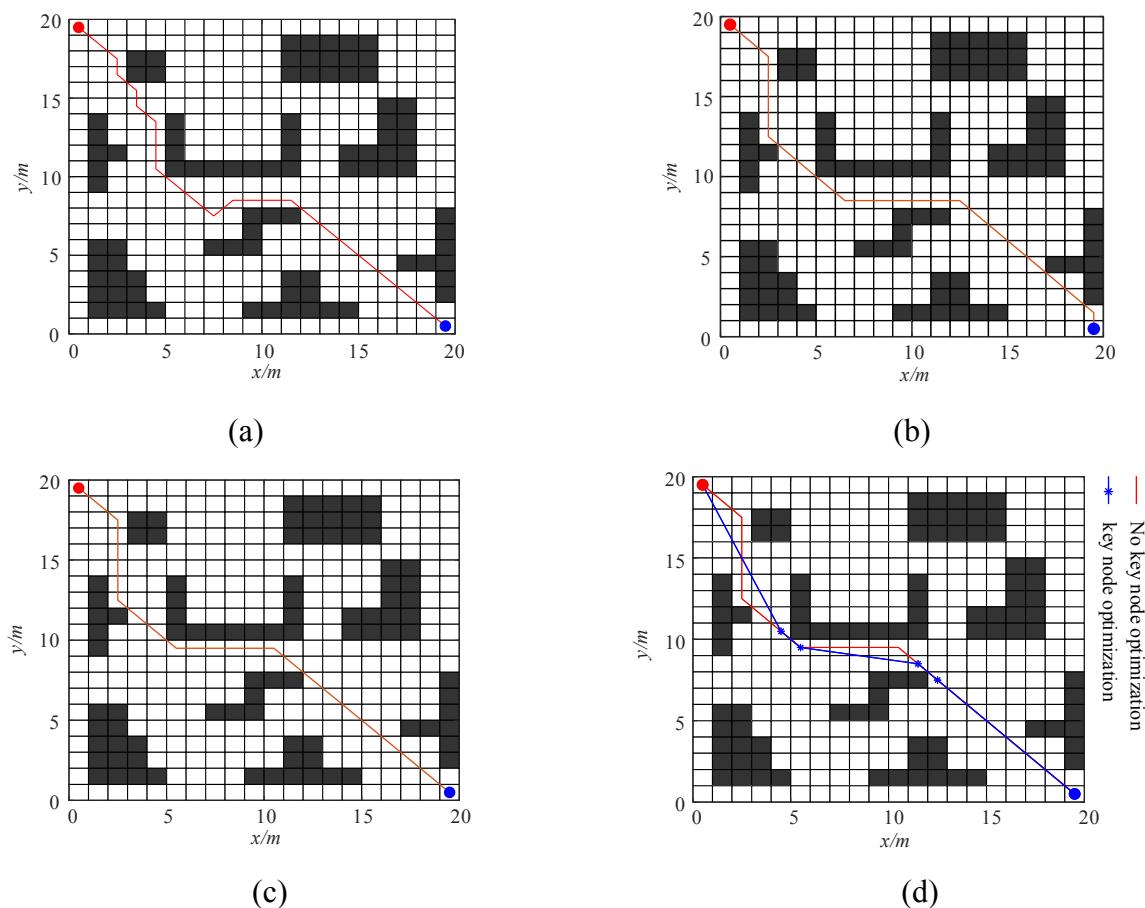


Figure 6. Comparison of the optimal paths for gradient experiments: (a) Experiment 1; (b) Experiment 2; (c) Experiment 3; (d) optimal path of the IACO.

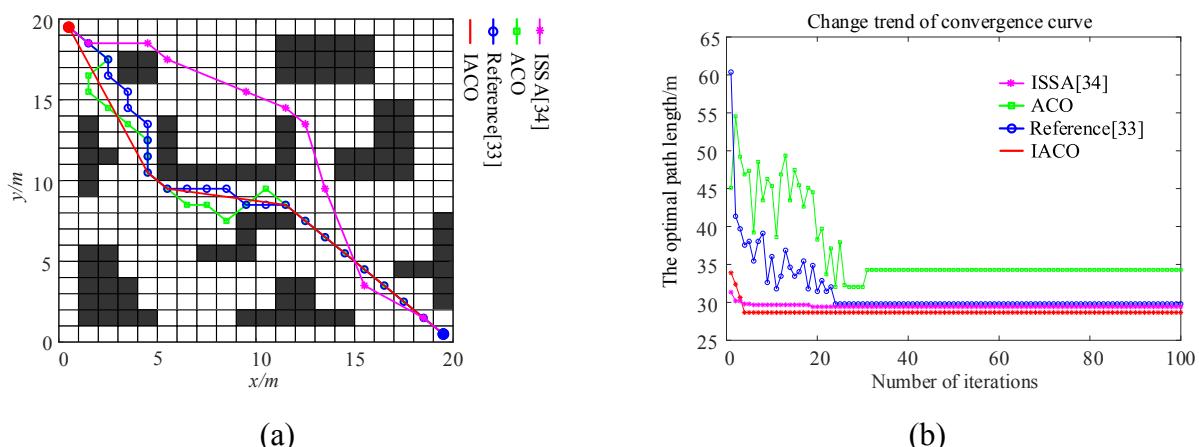


Figure 7. Comparison of the path-planning results for environment 1: (a) comparison of the optimal paths of the algorithms; (b) convergence comparison of the algorithms.

Let us assume that the starting point of the AGV is (0.5,19.5) and the target point is (19.5,0.5). The optimal path of the gradient experiment is shown in Figure 6; the optimal path of the four algorithms is shown in Figure 7(a); the algorithm convergence comparison is shown in Figure 7(b); and the gradient experiment results are compared in Table 1.

Table 1. Comparison of the gradient experiment results.

algorithm	Optimized path length/m			Number of turns			Number of iterations			Computing time/s		
	Best	Worst	Mean	Best	Worst	Mean	Best	Worst	Mean	Best	Worst	Mean
Experiment 1	30.6274	32.6274	31.4528	7	14	9	24	34	26	7.3519	8.1519	7.7435
Experiment 2	30.3848	32.0416	30.7660	5	9	5	21	30	22	8.1400	8.5116	8.4083
Experiment 3	29.7990	29.7990	29.7990	4	4	4	4	10	7	4.5592	5.4723	4.8123
ACO	32.0416	36.4558	32.9850	8	16	11	27	40	32	10.0538	12.2822	10.7681
Reference [33]	29.7990	31.2132	30.6542	6	12	9	24	41	34	8.5419	10.5290	9.2264
ISSA[34]	29.4183	29.9098	29.6048	7	10	9	20	50	29	6.5032	8.6980	6.9628
IACO	28.6595	28.6595	28.6595	3	3	3	4	9	6	8.2838	9.7868	8.8456

As revealed by the simulation results, although the traditional algorithm can obtain a collision-free path, the path quality and convergence speed need to be further improved, which is also a common problem of traditional ACO in complex maps.

For these problems, this paper improves the ACO from different aspects. The data show that after the addition of the improvement of Experiment 1, the path quality is improved, whether it is the optimal value or the worst value, which indicates that the adaptive pheromone strategy in this paper can prevent the ants from falling into the local optimal solution. Experiment 2 shows that after the addition of the improved heuristic function based on Experiment 1, the number of path turns is significantly reduced, which is the role of the improved heuristic function in this paper. Experiment 3 shows that after the addition of the pheromone improvement strategy in this paper based on Experiment 2, the convergence speed is significantly improved, and the overall path quality and overall algorithm search efficiency are higher. The pheromone update strategy proposed in this paper can help ants find better solutions and improve the convergence of the algorithm.

Figure 6(d) shows that the IACO shortens the path length and reduces the number of turns after using the key node optimization strategy to simplify the redundant inflection points, and the constructed path conforms to the movement rules of AGVs in the real environment. According to Table 1, compared with that of the traditional ACO and the algorithm in reference [33], the path length is reduced by 10.6 and 3.8%, the number of corners is reduced by 62.5 and 50.0%, and the number of iterations is reduced by 85.2 and 83.3%, respectively. In the simulation experiment, the optimal solution can still be obtained every time, which reflects the good stability of IACO in this paper. In addition, compared with the recently developed swarm intelligence algorithm [34], the path length is reduced by 2.6%, the number of corners is reduced by 57.1%, and the number of iterations is reduced by 80.0%. Compared with the other ant colony algorithms in Table 1, the computing time of the IACO is reduced, and the computing time increase is not obvious compared with ISSA.

6.2. Improved DWA algorithm experiment simulation analysis

In this section, environment 2 is selected for simulation experiments. Each experiment was run 20 times to verify the effectiveness of the improved DWA algorithm evaluation function and compared with the traditional DWA algorithm. Considering the radius of the real AGV, when the distance between AGV and target was less than 0.05 m, the AGV reached the target point. The experimental parameters of the IDWA in this paper were set as shown in Table 2.

Table 2. Parameter values of the IDWA.

parameter	Value
v_{\max}	1 m/s
ω_{\max}	20°/s
v_a	0.2 m/s ²
ω_a	50°/s ²
R	0.35 m
σ	0.2
τ	0.3
γ	0.2
ε	0.2
Linear Velocity Resolution	0.01 m/s
Angular velocity resolution	1°/s
Time resolution	0.1 s
Forecast period	3 s

6.2.1. Simulation experiment analysis of $Iodist(v, \omega)$

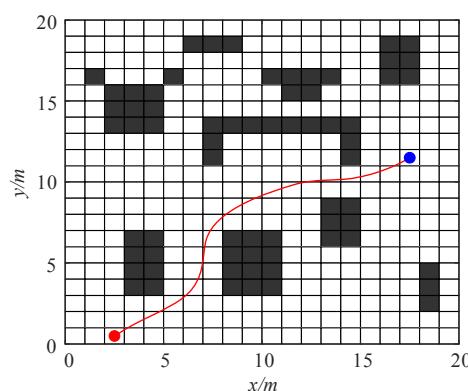


Figure 8. Path planned by the DWA after adding $Iodist(v, \omega)$.

Considering the unreasonable obstacle evaluation function mechanism of the DWA, it is easy for the AGV to detour and for the target to become unreachable. The experiment in this section selects the environment map in Figure 3 to verify the effectiveness of the improved obstacle distance function in this paper. Let us assume that the starting point of the AGV is (2.5,0.5) and the target point is (17.5,11.5). The planned path of the traditional DWA algorithm is shown in Figure 3, and the planned path of the DWA

algorithm after improving the obstacle evaluation function is shown in Figure 8.

In Figure 3, the path of the traditional DWA algorithm has a detour, which makes the target unreachable. In this paper, the obstacle distance evaluation function is improved. While ensuring safety, only obstacles within a certain range are considered. Figure 8 shows that the AGV can directly reach the target point through the free space among obstacles. The phenomenon of traditional DWA detours has been improved.

6.2.2. Simulation experiment analysis of $sgdist(v, \omega)$

Based on $Iodist(v, \omega)$, if the target point is near the obstacle, the phenomenon of the local optimal solution in Figure 4 will appear. The starting point of the AGV is $(2.5, 0.5)$, and the target point is $(9.5, 12.5)$. The traditional DWA algorithm planning path is shown in Figure 4, and the DWA algorithm planning path after adding the key node evaluation function is shown in Figure 9.

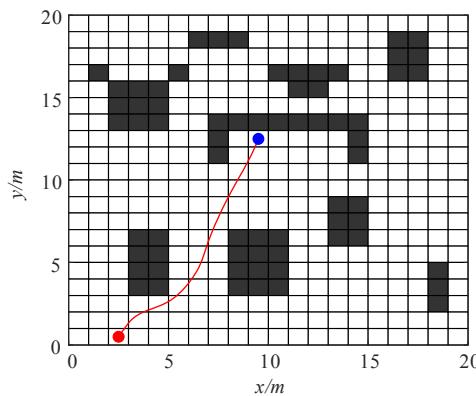


Figure 9. Path planned by the DWA after adding $sgdist(v, \omega)$.

Figures 4 and 9 show that after the key node evaluation function has been added, the AGV can more smoothly reach the destination. The results show that $sgdist(v, \omega)$ can enhance the ability of the AGV to move to the target point. Notably, most of the key nodes in this paper are inflection points near obstacles, which positively affect the navigation of the fusion algorithm to multiple local subtarget points.

6.3. Fusion algorithm simulation experiment analysis

6.3.1. 20×20 simulation environment

In a 20×20 small-scale complex environment, the starting point of the AGV is $(1.5, 0.5)$, and the target point is $(18.5, 18.5)$. The parameters of the IACO-DWA in this section are as follows: m is 50; NC is 100; β_{\max} and β_{\min} are 8.5 and 2.5, respectively; α_{\max} and α_{\min} are 1.5 and 0.5, respectively; ρ is 0.3; Q is 1; $q_0 \in (0, 1)$; q_0 is 0.9; a and b are 0.7 and 0.3; R is 0.5; v_{\max} is 1 m/s; ω_{\max} is $20^\circ/\text{s}$; v_a is 0.2 m/s; ω_a is $50^\circ/\text{s}$; R is 0.35 m; σ is 0.2; τ is 0.3; γ is 0.2; ε is 0.2; the linear velocity resolution is 0.01 m/s; the angular velocity resolution is $1^\circ/\text{s}$; the time

resolution is 0.01 s; the forecast period is 3 s. The parameter settings in reference [35] are as follows: v_{\max} is 1 m/s; ω_{\max} is $20^{\circ}/s$; v_a is 0.2 m/s; ω_a is $50^{\circ}/s$; R is 0.35 m; σ is 0.3; τ is 0.05; γ is 0.2; the linear velocity resolution is 0.01 m/s; and the angular velocity resolution is $1^{\circ}/s$. To reduce the accidental nature of the experiment, each algorithm was performed 20 times, and the algorithm performance comparison table was drawn for comprehensive evaluation. All values in Table 3 represent the optimal value of the 20 times. The experimental results are shown in Figure 10. The dashed line in Figure 10(b) indicates the global path planned by the IACO, and the asterisks on the dashed line indicate the key nodes. The performance table of the path evaluation index is shown in Table 3.

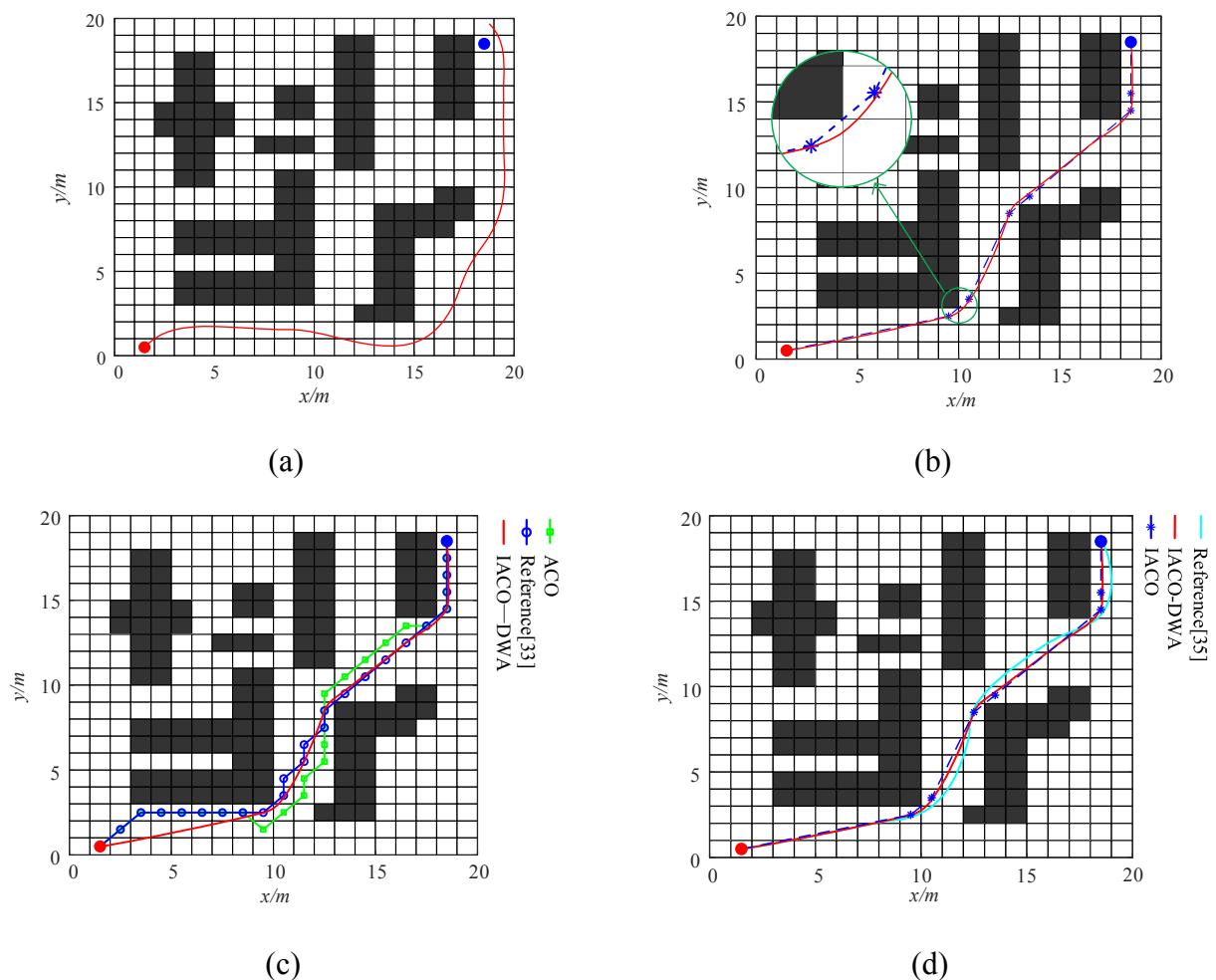


Figure 10. Comparison of the optimal path in a 20×20 environment: (a) path planned by DWA; (b) path planned by the IACO-DWA; (c) optimal path of IACO-DWA compared to reference [33] and ACO; (d) optimal path of IACO-DWA compared to reference [35].

The experimental results show that the IACO-DWA solves the following problem: the traditional DWA algorithm lacks global guidance, the path is long, and the planned path cannot reach the target point. In addition, the IACO-DWA can better track the global path and basically guarantees the global optimality of the path. As shown in Figure 10(b), the IACO-DWA satisfies the AGV motion characteristics, which improves the smoothness of the path and ensures the safety of the AGV during

movement. In Table 3, compared with the traditional ACO and the algorithm in [33], IACO-DWA reduces the path length by 9.9 and 3.5% and the number of turns by 60 and 56.6%, respectively, while ensuring safety and smoothness. As shown in Figure 10(d), the fusion algorithm in [33] gradually deviates from the global path planned by IACO in the process of tracking subtarget points because it cannot track local subtargets. Because the fusion algorithm in this paper enhances the ability of the DWA algorithm to track the target point, it can better track the global path. Compared with the existing fusion algorithm, the path length is reduced by 3.1%, and the overall path quality is better than that of the algorithm proposed in [35]. In summary, the IACO-DWA fully utilizes the advantages of the two path planning algorithms, avoids their shortcomings in path planning, achieves simultaneous path optimization with multiple objectives (path length, number of turns, safety and smoothness) and improves the planning efficiency.

Table 3. Comparison of 20×20 experimental results.

Algorithm	Whether to reach the target point/Optimal path length (m)	Number of turns	Smoothness	Safety
DWA	No	-	Yes	Yes
ACO	30.5563	10	No	Worst
Reference [33]	28.5563	9	No	Worst
Reference [35]	28.4260	4	Yes	Yes
IACO-DWA	27.5460	4	Yes	Yes

6.3.2. 50×50 simulation environment

In the actual environment, the working space of the AGV is often a large-scale environment. Therefore, this paper uses a 50×50 scale grid map and randomly places complex obstacles for the simulation experiments. The experimental parameters in this section are identical to the experimental parameter settings in Section 6.3.1 and will not be repeated here. The starting point of the AGV is $(3.5, 0.5)$, and the target point is $(47.5, 45.5)$. The path comparison results between the IACO-DWA and other algorithms are shown in Figure 11, and the path evaluation index performance table is shown in Table 4.

As shown by point “A” in Figure 10(a), in a large-scale complex environment, the traditional DWA algorithm is prone to the phenomenon that the target is unreachable. Since the fusion algorithm in this paper makes full use of the global information of the path planned by the IACO and improves the obstacle evaluation function of the DWA algorithm, it increases the local optimization ability of DWA. The experimental results fully prove that the fusion algorithm in this paper solves the problem of falling into local optimal solutions. Compared with the traditional ACO and the algorithm in [33], the path length of IACO-DWA planning is reduced by 14.1 and 3.8%, respectively, the number of turns is reduced by 54.8 and 30.0%, respectively, and the path is safer and smoother. As shown in Figure 11(d), compared with the algorithm in reference [35], the path planned by IACO-DWA is shorter, smoother, and higher in path quality. The experimental results fully show that in a large-scale and complex environment, the IACO-DWA retains its advantages. Compared with the 20×20 scale map experiment, when the map size increases, the fusion algorithm in this paper also plans a high-quality path that simultaneously satisfies multiple path optimization objectives.

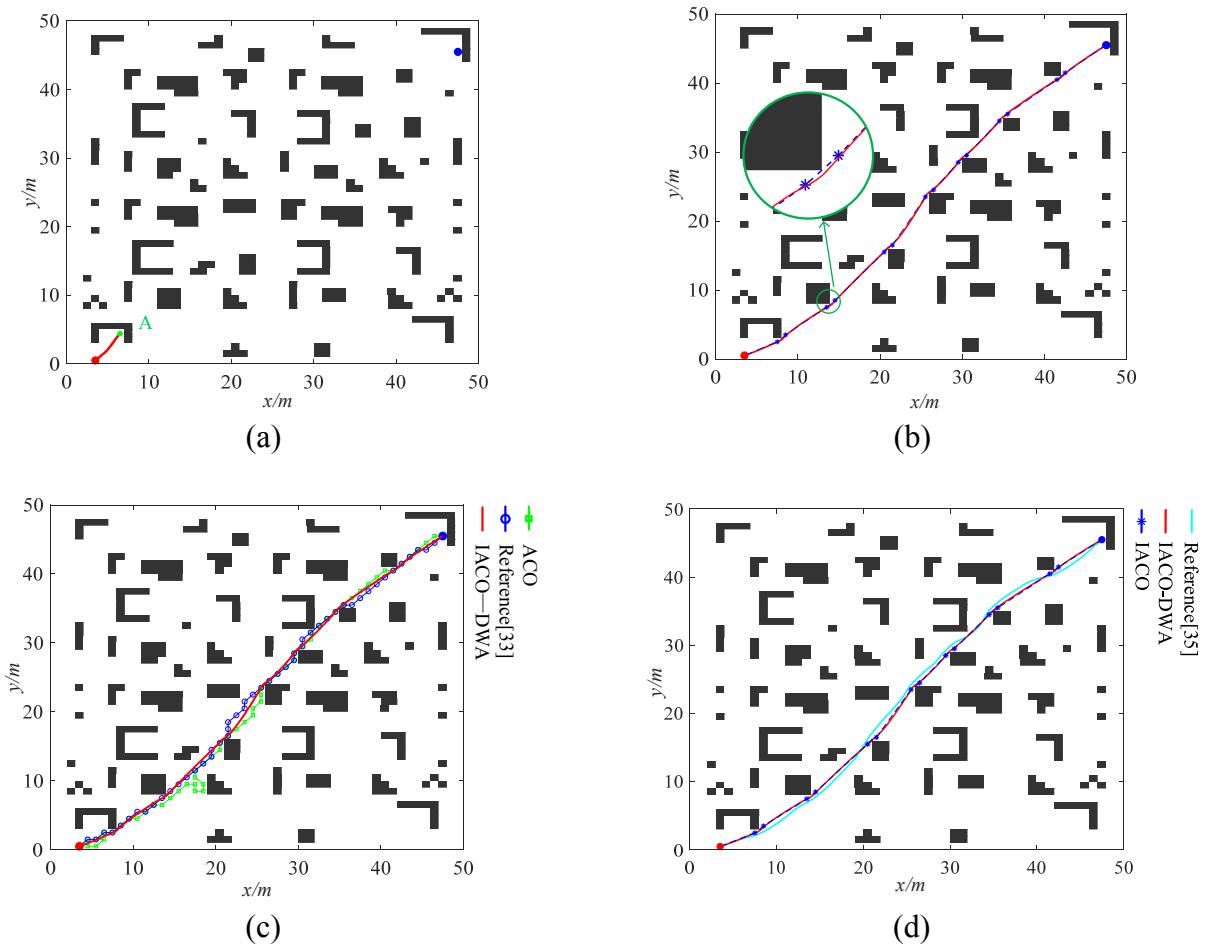


Figure 11. Comparison of the optimal path in the 50×50 environment: (a) path planned by the traditional DWA algorithm; (b) path planned by IACO-DWA; (c) optimal path of IACO-DWA compared to reference [33] and ACO; (d) optimal path of IACO-DWA compared to reference [35].

Table 4. Comparison of 50×50 experimental results.

Algorithm	Whether to reach the target point/Optimal path length (m)	Number of turns	Smoothness	Safety
DWA	No	-	Yes	Yes
ACO	74.1543	31	No	Worst
Reference [33]	66.1573	20	No	Worst
Reference [35]	64.7160	14	Yes	Yes
IACO-DWA	63.6731	14	Yes	Yes

6.4. Experimental verification

The algorithm in this paper is applied to the second-generation nonholonomic wheeled mobile

robot QBot2e produced by Quanser company in Canada to verify its effectiveness and feasibility in the actual indoor environment.

In this paper, an indoor laboratory was selected as the experimental environment, and multiple obstacles were placed indoors to conduct path planning experiments. The movement of the AGV was controlled, and the QBot2e vehicle-mounted Kinect camera was used to obtain the external environment information, which can generate the depth information of the environment. This information combined with the position and orientation of the AGV chassis can be employed for autonomous map construction, as shown in Figure 12. Using image processing technology, the depth information containing the environment was processed into a two-dimensional grid map required for AGV path planning. The four algorithms were applied to the indoor environment, and the indoor real experimental algorithm parameters were consistent with the computer algorithm simulation parameters. The optimal path is shown in Figure 13.



Figure 12. Indoor actual environment.

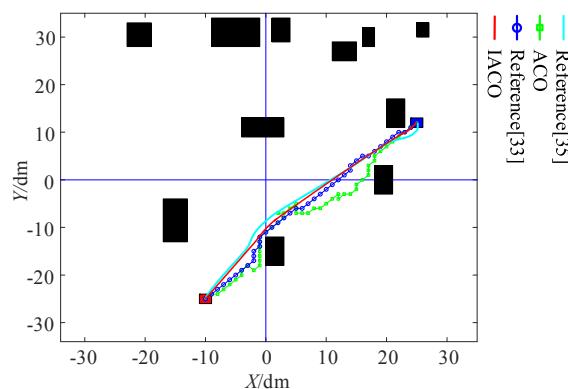


Figure 13. Comparison of optimal paths in an actual environment.

Figure 13 shows that the IACO-DWA planned an obviously better path than the traditional algorithm and other existing methods [35], which fully demonstrates the superiority of the proposed algorithm in this paper. The target path points generated by the path planning algorithm were used as vectors and separately sent to the AGV. Then, the path point was compared with the current position of the AGV; the PID controller was utilized in the feedback loop to control the movement of the AGV until the AGV reached the target point, and the algorithm parameters were consistent with the computer simulation parameters. The indoor test results are shown in Figure 14.

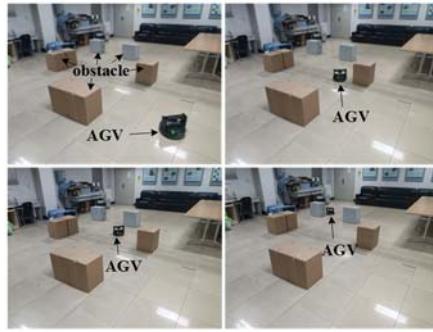


Figure 14. Indoor experimental results.

Considering the small friction on the ground in the laboratory, the AGV will cause positional deviation due to wheel slippage during the movement process. Therefore, a certain limit is set for the speed of the AGV, which does not affect the overall path-planning and path-tracking process. The indoor experimental results show that the AGV moves according to the path planned by the IACO-DWA, successfully avoids the obstacle, and maintains a certain safe distance from the obstacle. When the algorithm in this paper was run 10 times on the map, the trajectory remained almost constant, and the path at the corner was very smooth, which satisfied the motion constraints of the actual AGV and reduced the wear on the body hardware caused by the discontinuity of turning. These indoor experiment results verify the practicability of the proposed algorithm in indoor AGV path planning.

7. Conclusions

In this paper, in large-scale or complex environments, the traditional ACO plans a large number of path turns, the convergence speed is slow, and the traditional DWA has problems such as detours and unreachable target points. An efficient path-planning algorithm that combines ACO and DWA is proposed. To address the shortcomings of the two algorithms in path planning, a corresponding optimization scheme is proposed, which avoids their own limitations and simultaneously optimizes multiple path indicators: the path length, number of turns, smoothness, and safety. By comparing two maps of different scales for simulation experiments and from a multi-angle analysis of the algorithm's path-planning evaluation performance indicators, the effectiveness and superiority of the fusion algorithm in this paper are verified. The algorithm of this paper is transplanted to the QBot2e mobile robot produced by the Quanser company, and the experimental results are consistent with the simulation experiments, which verifies the practicability and superiority of the fusion algorithm in this paper. In summary, the algorithm in this paper can be applied to indoor AGV path planning in large-scale and complex environments and has a certain practical value. This paper is mainly for indoor AGV global path planning. In the next step, we will consider combining sensors and local obstacle avoidance algorithms to conduct research on dynamic obstacle avoidance and multi-AGV coordinated path planning.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62103127), Natural Science Foundation of Hebei Province of China (F2020201048) and the Central Guidance on Local Science and Technology Development Fund of Hebei Province (19941822G).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. M. De Ryck, M. Versteyhe, F. Debrouwere, Automated guided vehicle systems, state-of-the-art control algorithms and techniques, *J. Manuf. Syst.*, **54** (2020), 152–173. <https://doi.org/10.1016/j.jmsy.2019.12.002>
2. K. Akka, F. Khaber, Mobile robot path planning using an improved ant colony optimization, *Int. J. Adv. Robot. Syst.*, **15** (2018). <https://doi.org/10.1177/1729881418774673>
3. B. Patle, A. Pandey, D. Parhi, A. Jagadeesh, A review: on path planning strategies for navigation of mobile robot, *Def. Technol.*, **15** (2019), 582–606. <https://doi.org/10.1016/j.dt.2019.04.011>
4. 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>
5. 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>
6. X. Xiong, H. Min, Y. Yu, P. Wang, Application improvement of A* algorithm in intelligent vehicle trajectory planning, *Math. Biosci. Eng.*, **18** (2021), 1–21. <https://doi.org/10.3934/mbe.2021001>
7. A. Ammar, H. Bennaceur, I. Châari, A. Koubâa, M. Alajlan, Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments, *Soft Comput.*, **20** (2016), 4149–4171. <https://doi.org/10.1007/s00500-015-1750-1>
8. C. Xia, Y. Zhang, I. Chen, Learning sampling distribution for motion planning with local reconstruction-based self-organizing incremental neural network, *Neural. Comput. Appl.*, **31** (2019), 9185–9205. <https://doi.org/10.1007/s00521-019-04370-y>
9. R. K. Dewangan, A. Shukla, W. W. Godfrey, Three dimensional path planning using Grey wolf optimizer for UAVs, *Appl. Intell.*, **49** (2019), 2201–2217. <https://doi.org/10.1007/s10489-018-1384-y>
10. L. Zhang, Y. Zhang, Y. Li, Mobile robot path planning based on improved localized particle swarm optimization, *IEEE Sens. J.*, **21** (2020), 6962–6972. <https://doi.org/10.1109/JSEN.2020.3039275>
11. 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>
12. X. Li, L. Wang, L. Wang, Application of improved ant colony optimization in mobile robot trajectory planning, *Math. Biosci. Eng.*, **17** (2020), 6756–6774. <https://doi.org/10.3934/mbe.2020352>
13. H. Yang, J. Qi, Y. Miao, H. Sun, J. Li, A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization, *IEEE Trans. Ind. Electron.*, **66** (2019), 8557–8566. <https://doi.org/10.1109/TIE.2018.2886798>
14. Y. Zheng, Q. Luo, H. Wang, C. Wang, X. Chen, Path planning of mobile robot based on adaptive ant colony algorithm, *J. Intell. Fuzzy Syst.*, **39** (2020), 5329–5338. <https://doi.org/10.3233/JIFS-189018>

15. J. Chen, Y. Zhang, L. Wu, T. You, X. Ning, An adaptive clustering-based algorithm for automatic path planning of heterogeneous UAVs, *IEEE Trans. Intell. Transp. Syst.*, **2021** (2021), 1–12. <https://doi.org/10.1109/TITS.2021.3131473>
16. J. Chen, C. Du, Y. Zhang, P. Han, W. Wei, A clustering-based coverage path planning method for autonomous heterogeneous UAVs, *IEEE Trans. Intell. Transp. Syst.*, **2021** (2021), 1–11. <https://doi.org/10.1109/TITS.2021.3066240>
17. Z. Jiao, K. Ma, Y. Rong, H. Zhang, S. Wang, A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs, *J. Comput. Sci.*, **25** (2018), 50–57. <https://doi.org/10.1016/j.jocs.2018.02.004>
18. J. Chen, F. Ling, Y. Zhang, T. You, Y. Liu, X. Du, Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system, *Swarm Evol. Comput.*, **69** (2022), 101005. <https://doi.org/10.1016/j.swevo.2021.101005>
19. S. Kumar, D. R. Parhi, M. K. Muni, K. K. Pandey, Optimal path search and control of mobile robot using hybridized sine-cosine algorithm and ant colony optimization technique, *Ind. Rob.*, **47** (2020), 535–545. <https://doi.org/10.1108/IR-12-2019-0248>
20. W. Gao, Q. Tang, B. Ye, Y. Yang, J. Yao, An enhanced heuristic ant colony optimization for mobile robot path planning, *Soft Comput.*, **24** (2020), 6139–6150. <https://doi.org/10.1007/s00500-020-04749-3>
21. Z. Zhou, J. Wang, Z. Zhu, D. Yang, J. Wu, Tangent navigated robot path planning strategy using particle swarm optimized artificial potential field, *Optik*, **158** (2018), 639–651. <https://doi.org/10.1016/j.ijleo.2017.12.169>
22. E. J. Molinos, A. Llamazares, M. Ocaña, Dynamic window based approaches for avoiding obstacles in moving, *Rob. Auton. Syst.*, **118** (2019), 112–130. <https://doi.org/10.1016/j.robot.2019.05.003>
23. X. Bai, H. Jiang, J. Cui, K. Lu, P. Chen, M. Zhang, UAV path planning based on improved A* and DWA algorithms, *Int. J. Aerosp. Eng.*, **2021** (2021). <https://doi.org/10.1155/2021/4511252>
24. 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>
25. Z. Lin, M. Yue, G. Chen, J. Sun, Path planning of mobile robot with PSO-based APF and fuzzy-based DWA subject to moving obstacles, *Trans. Inst. Meas. Control.*, **44** (2022), 121–132. <https://doi.org/10.1177/01423312211024798>
26. X. Li, F. Liu, J. Liu, S. Liang, Obstacle avoidance for mobile robot based on improved dynamic window approach, *Turk. J. Electr. Eng. Comput. Sci.*, **25** (2017), 666–676. <https://doi.org/10.3906/elk-1504-194>
27. R. J. Mullen, D. Monekosso, S. Barman, P. Remagnino, A review of ant algorithms, *Expert Syst. Appl.*, **36** (2009), 9608–9617. <https://doi.org/10.1016/j.eswa.2009.01.020>
28. Q. Luo, H. Wang, Y. Zheng, J. He, Research on path planning of mobile robot based on improved ant colony algorithm, *Neural Comput. Appl.*, **32** (2020), 1555–1566. <https://doi.org/10.1007/s00521-019-04172-2>
29. W. Li, L. Xia, Y. Huang, S. Mahmoodi, An ant colony optimization algorithm with adaptive greedy strategy to optimize path problems, *J. Ambient Intell. Hum. Comput.*, **13** (2022), 1557–1571. <https://doi.org/10.1007/s12652-021-03120-0>

30. S. Li, X. You, S. Liu, Co-evolutionary multi-colony ant colony optimization based on adaptive guidance mechanism and its application, *Arabian J. Sci. Eng.*, **46** (2021), 9045–9063. <https://doi.org/10.1007/s13369-021-05694-5>
31. D. Lee, S. Lee, C. Ahn, C. Lim, Finite distribution estimation-based dynamic window approach to reliable obstacle avoidance of mobile robot, *IEEE Trans. Ind. Electron.*, **68** (2020), 9998–10006. <https://doi.org/10.1109/TIE.2020.3020024>
32. 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>
33. X. You, S. Liu, C. Zhang, An improved ant colony system algorithm for robot path planning and performance analysis, *Int. J. Robot. Autom.*, **33** (2018), 527–533. <https://doi.org/10.2316/Journal.206.2018.5.206-0071>
34. Z. Zhang, R. He, K. Yang, A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm, *Adv. Manuf.*, **10** (2022), 114–130. <https://doi.org/10.1007/s40436-021-00366-x>
35. X. Chi, H. Li, J. Fei, Research on random obstacle avoidance method for robots based on the fusion of improved A[~]* algorithm and dynamic window method, *Chin. J. Sci. Instrum.*, **42** (2021), 132–140. <https://doi.org/10.19650/j.cnki.cjsi.J2007064>



©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)