Mathematical Biosciences and Engineering

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

MBE, 21(2): 2189–2211.
DOI: 10.3934/mbe.2024096
Received: 29 November 2023
Revised: 21 December 2023
Accepted: 04 January 2024
Published: 11 January 2024

Research article

A new path planning strategy integrating improved ACO and DWA algorithms for mobile robots in dynamic environments

Baoye Song*, **Shumin Tang and Yao Li**

College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China

* **Correspondence:** Email: songbaoye@sdust.edu.cn.

Abstract: This article is concerned with the path planning of mobile robots in dynamic environments. A new path planning strategy is proposed by integrating the improved ant colony optimization (ACO) and dynamic window approach (DWA) algorithms. An improved ACO is developed to produce a globally optimal path for mobile robots in static environments. Through improvements in the initialization of pheromones, heuristic function, and updating of pheromones, the improved ACO can lead to a shorter path with fewer turning points in fewer iterations. Based on the globally optimal path, a modified DWA is presented for the path planning of mobile robots in dynamic environments. By deleting the redundant nodes, optimizing the initial orientation, and improving the evaluation function, the modified DWA can result in a more efficient path for mobile robots to avoid moving obstacles. Some simulations are conducted in different environments, which confirm the effectiveness and superiority of the proposed path planning algorithms.

Keywords: mobile robots; path planning; ant colony optimization; dynamic window approach; ACO; DWA

1. Introduction

Path planning is to find the optimal path from a starting point to a destination in an environment with obstacles [1, 2]. Up to now, path planning has been used in many applications, such as robot navigation, autonomous driving, logistics transportation, and unmanned aerial vehicles [3,4]. Through path planning, mobile robots or autonomous vehicles can find safe and efficient paths for their tasks in complex environments.

The path planning of mobile robots can be divided into two categories: global path planning in static environments, and local path planning in dynamic environments [5]. To date, a large number of algorithms have been developed for the path planning of mobile robots, e.g., the dijkstra algorithm [6], A* algorithm [7], artificial potential field (APF) algorithm [8], fuzzy system [9], ant colony optimization (ACO) [10], particle swarm optimization (PSO) [11], and dynamic window approach (DWA) [12]. Among others, DWA is a local obstacle avoidance method based on velocity space, which considers the mobile robot's kinematic and dynamic constraints [13]. By exploring a range of linear and angular velocities satisfying the constraints, a set of optimal velocities for controlling the robot's motion are selected based on a velocity evaluation function [14]. Due to its simplicity, smooth path, and strong obstacle-avoidance capabilities, DWA has been widely applied in mobile robot path planning in recent years [3].

Nevertheless, traditional DWA still has some drawbacks, such as low algorithm efficiency in dynamic obstacle environments, especially when the target points are far from the starting points [15]. In some complex scenes, it is prone to getting trapped in local optima or even failing to reach the destination [16]. To address these problems, researchers have proposed many improved algorithms. For example, an improved DWA has been presented in [17], where the obtained paths can be optimized for obstacle avoidance, but it is limited to local path planning. In [18], the path generated by DWA is smooth and exhibits obstacle-avoidance capabilities, but this algorithm is prone to getting trapped in deadlocks in some cases. In [19], an adaptive dynamic adjustment factor has been introduced into the velocity evaluation function of the traditional DWA. By taking into account the density of obstacles in front of the mobile robot, this method can improve the performance of traditional DWA when navigating through dense obstacles, yet the effectiveness of this method has not been validated in dynamic environments. Additionally, it is worth noting that global paths are seldom considered in the aforementioned DWA-based mobile robot path planning algorithms.

To overcome the above-mentioned drawbacks, in recent years, the hybrid algorithms combining DWA have received extensive research and attention in the field of dynamic path planning [20]. In particular, the integration of DWA with global path planning algorithms allows for escaping local optima and successfully reaching the target by using the guidance provided by globally optimal paths during local path planning [21]. For example, an algorithm combining DWA with an improved A* algorithm has been developed for the path planning of greenhouse robots [22]. The fusion of the improved genetic algorithm (GA) with DWA has been presented for local real-time smooth path planning of robots [23]. The combination of ACO and DWA has improved the evaluation function of DWA, resulting in enhanced path-tracking capability, dynamic obstacle avoidance ability, and motion stability of robots, thus attracting more attention from researchers [24].

In the hybrid algorithms combining DWA, ACO has many advantages for the path planning of mobile robots, such as self-organization, positive feedback, parallel computing, and robustness [25]. Nevertheless, traditional ACO also has some limitations, such as the susceptibility to getting stuck in local optima and having many turning points in resultant paths [26]. To combat these deficiencies, several improvements have been proposed for ACO-based path planning algorithms. For example, the state transition rules of ACO have been improved to enhance the convergence speed of the algorithm [27], but it still faces the issue of blindly searching due to the uniformly distributed initial state of pheromones. In [28], an adaptive greedy strategy has been used to accelerate the convergence speed of ACO. Though the adaptively adjusted parameters can increase the diversity of the population, the algorithm has to overcome the challenging problem of getting out of the local optima. Hence, there is still much room to further improve the performance of the ACO algorithm.

Motivated by the aforementioned discussions, in this article, we propose a new path planning strategy for mobile robots in dynamic environments based on improved ACO and DWA algorithms. An improved ACO algorithm is developed for the global path planning of mobile robots in static environments. Based on the obtained global path, an improved DWA algorithm is proposed to design an optimal path for mobile robots in dynamic environments. The effectiveness and superiority of the new strategy integrating the improved ACO and DWA algorithms are confirmed through some simulations on the path planning of mobile robots. The main contributions of this article can be summarized as follows:

1) A path planning strategy that combines the improved ACO and DWA algorithms is proposed for mobile robots in dynamic environments;

2) An improved ACO is developed to obtain a globally optimal path through improvements in the initialization of pheromones, heuristic function, and updating of pheromones;

3) A modified DWA is presented for path planning of mobile robots in dynamic environments based on the generated globally optimal path by deleting the redundant nodes, optimizing the initial orientation, and improving the evaluation function;

4) Some simulation experiments are carried out for the path planning of mobile robots in different environments, and the effectiveness and superiority of the new algorithms are verified by the simulation results.

The remainder of this article is outlined as follows. Section 2 presents some preliminary information on the path planning of mobile robots. The global path planning based on the improved ACO algorithm is detailed in Section 3. Section 4 is devoted to the dynamic path planning based on the improved DWA algorithm. Several simulation results and discussions are reported in Section 5. Finally, the article is concluded in Section 6.

2. Preliminaries

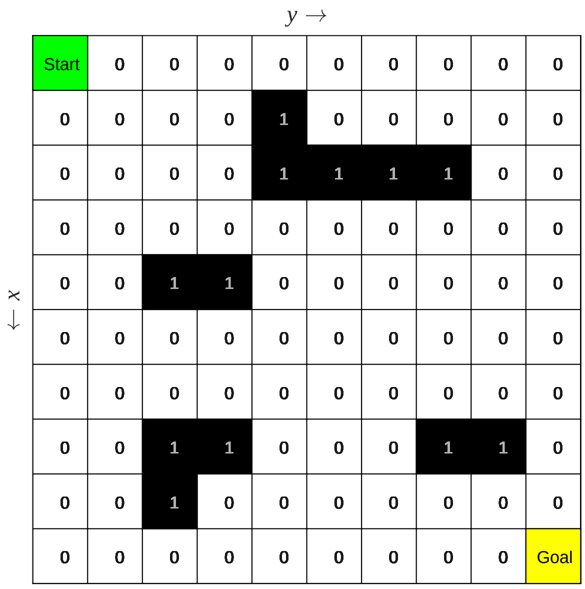
2.1. Workspace model

An appropriate workspace model is beneficial for achieving satisfactory path planning results. Commonly used workspace modeling methods include visibility graph methods, topological methods, free-space planning methods, and grid-based methods [3]. In contrast, the grid-based method stands out due to its simplicity in modeling and convenient computation [29], which motives us to use it in the current study.

As an example, we examine the grid map depicted in Figure 1, where "1" indicates impassable regions displayed as black cells, and "0" denotes accessible areas represented by white cells. Each grid in the workspace is numbered in order from left to right and top to bottom. The position of each grid is represented by the coordinates (x, y) of its center point, where x and y represent the vertical and horizontal directions, respectively. The coordinates of the starting grid in the top left corner are defined as $(1, 1)$, and the target grid's coordinates in the bottom right corner are defined as $(10, 10)$. The conversion rule between the coordinates of each grid and the corresponding number is defined as follows

$$\begin{cases} x = \text{floor}\left(\dfrac{n-1}{N_x}\right) + 1, \\ y = \text{mod}(n-1, N_y) + 1 \end{cases} \tag{2.1}$$

where floor(·) is the rounding down operation; mod(·) is the modulo operation; n denotes the number of the grid; and N_x and N_y (both are 10 in Figure 1) indicate the amounts of grids in columns and rows, respectively. Additionally, in the path planning of this article, the mobile robot is taken as a point in the workspace without considering the actual size.



Start	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	Goal

Figure 1. An example for the grid map of a workspace.

2.2. A algorithm*

A* algorithm is a heuristic search algorithm widely used for the path planning of mobile robots [30]. By using the evaluation function to select the optimal nodes, the A* algorithm can improve the efficiency of finding the shortest path in a grid-based workspace.

The evaluation function used to estimate the quality of each node includes two factors [22], i.e., $g(n)$ and $h(n)$. By adding two factors together, a comprehensive cost function can be obtained as

$$f(n) = g(n) + h(n) \tag{2.2}$$

where n denotes the current node's number; $g(n)$ indicates the actual cost of moving from the starting node to the current node; $h(n)$ represents the predicted cost of moving from the current node to the target node, which is also called the heuristic function; and $f(n)$ is the total cost of moving from the starting point to the endpoint through the current node. In this article, we adopt the Euclidean distance to calculate the heuristic function that is expressed by:

$$h(n) = \sqrt{(x_n - x_t)^2 + (y_n - y_t)^2} \tag{2.3}$$

where (x_n, y_n) and (x_t, y_t) are the coordinates of the current and target nodes, respectively.

3. Global path planning based on improved ACO

3.1. Traditional ACO algorithm

The ACO algorithm is an intelligent optimization algorithm that simulates the foraging behavior of ant colonies [31]. Ant colonies can always find the optimal path from the nest to the food source because each ant leaves behind a chemical substance called pheromone during the foraging process. Pheromones gradually evaporate over time, so the pheromone concentration is higher on the shorter paths since there are more ants on the paths, leaving more pheromones. Other ants can sense the pheromone concentration and follow the paths with higher pheromone concentrations, thereby further reinforcing the pheromone concentration on the paths.

The ACO algorithm abstracts the foraging behavior of ants into a mathematical model choosing paths based on a state transition probability [32], which is expressed as

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum\limits_{l \in G_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}(t)]^\beta}, & j \in G_i^k \\ 0, & \text{others} \end{cases} \tag{3.1}$$

where $P_{ij}^k(t)$ denotes the probability for the k-th ant walking on the path from the i-th grid to the j-th grid (denoted as $path(i, j)$) at the t-th iteration; the corresponding concentration of pheromones on $path(i, j)$ is denoted as $\tau_{ij}(t)$, which represents the influence of the search history on the path choice of the ants, where α denotes the importance factor associated with the pheromone concentration; $\eta_{ij}(t)$ is a heuristic function representing the attraction of the j-th grid for ants at the t-th iteration, where β denotes the importance factor associated with the attraction of the grids; and G_i^k indicates the set of grids that have not been explored by the k-th ant when it is currently at the i-th grid. Generally, the heuristic function $\eta_{ij}(t)$ is defined by

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \tag{3.2}$$

where d_{ij} denotes the Euclidean distance between the i-th and j-th grids. Meanwhile, the pheromone concentration will be updated during each iteration as follows:

$$\begin{cases} \tau_{ij}(t+1) = (1-\lambda)\tau_{ij}(t) + \Delta\tau_{ij}(t), \\ \Delta\tau_{ij}(t) = \sum\limits_{k=1}^{N} \Delta\tau_{ij}^k(t), \end{cases} \tag{3.3}$$

where λ is the pheromone diffusion factor, $\Delta\tau_{ij}$ is the increment of pheromones on $path(i, j)$, N denotes the number of ants, and $\Delta\tau_{ij}^k$ represents the released pheromone by the k-th ant on $path(i, j)$ and can be defined by

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k, & \text{if the ant passed } path(i, j) \\ 0, & \text{others} \end{cases} \tag{3.4}$$

where Q is the strength of pheromones, and L_k denotes the path and its length of the k-th ant.

Remark 1. *From the above introduction of the traditional ACO algorithm, it can be observed that the pheromone on a path is negatively correlated with the path length, i.e., the shorter path retains more* *pheromones. For paths with higher pheromone values, they have a higher probability of being chosen by ants in the next iteration, thereby further increasing the pheromone concentration retained on the paths. Through the positive feedback mechanism of the pheromones, ants can gradually converge on better paths and ultimately find the optimal solution to the path planning problem.*

3.2. Improved ACO algorithm

In this article, the improvement of the ACO algorithm focuses on the following three aspects: initialization of pheromones, heuristic function, and updating of pheromones, which will be discussed in the remainder of this section.

3.2.1. Initialization of pheromones

In traditional ACO, each path has the same initial pheromone concentration, which would lead to a slow convergence speed in the search process [33]. To address this issue, the A* algorithm is employed to initialize the pheromone concentration, aiming to accelerate the convergence speed of ACO and enhance its search performance.

In the new initialization scheme, the A* algorithm is used to obtain the initial path L_0 (see the red dotted path in Figure 2 for example) from the starting grid to the target grid. To prevent the algorithm from excessively relying on the guidance of the initial path and getting trapped in local minima, we construct a candidate region R (see the green region in Figure 2) that is composed of the path L_0 and its eight neighboring grids. In the region R, the pheromone concentration between the i-th and j-th grids is initialized as

$$\tau_{ij} = \begin{cases} \rho \cdot \tau_{\text{init}}, & i, j \in R \\ \tau_{\text{init}}, & \text{others} \end{cases} \tag{3.5}$$

where $\rho > 1$ is a factor for grids in the region R, and τ_{init} is the initial pheromone concentration for the grids out of the region R.

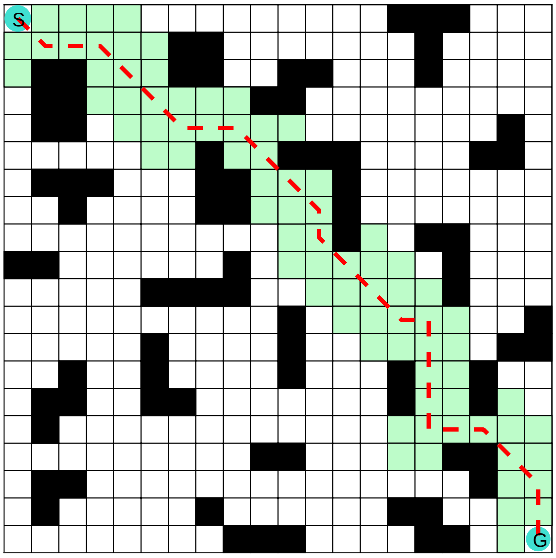


Figure 2. An example for the initialization of pheromones.

3.2.2. Heuristic function

In traditional ACO, the heuristic function $\eta_{ij}(t)$ (see, e.g., Eq (3.2)) only involves the distance from the current i-th grid to the next selectable j-th grid, which would lead to a lack of guidance in the ACO's search process, resulting in slow convergence or getting trapped in local optima [34]. To overcome this limitation, the distance between the next selectable grid j and the target grid t is integrated into the heuristic function, which is redefined as

$$\eta_{ij}(t) = \frac{1}{\sigma d_{ij} + (1-\sigma) d_{jt}} \tag{3.6}$$

where $\sigma \in (0, 1)$ is a weighting parameter of two distances, d_{ij} is the Euclidean distance between the i-th and j-th grids, and d_{jt} is the Euclidean distance from the j-th grid to the target grid.

3.2.3. Updating of pheromones

The pheromone is one of the important factors influencing the path selection of ants. The updating of pheromones is composed of pheromone accumulation and pheromone evaporation. In each iteration, we refer to the ant that has traveled the shortest path as the "best ant", and similarly, the ant that has traveled the longest path is called the "worst ant" [26]. The pheromones are first updated according to Eq (3.3). Then, the shortest and longest paths are identified among all paths, and a reward and a punishment are respectively given to the pheromones on the best and worst paths, in order to enhance the guiding effect of the optimal solution for subsequent iterations and reduce the misleading effect of the worst solution in future iterations. Thus, combining the traditional ant colony algorithm with the above best-worst ant system, the additional pheromone increments for the best and worst paths are designed as

$$\begin{cases} \Delta\tau_{ij}^{b}(t) = \dfrac{Q}{L_b}, \\ \Delta\tau_{ij}^{w}(t) = -\dfrac{Q}{L_w}, \end{cases} \tag{3.7}$$

where L_b and L_w denote the length of the best (shortest) and worst (longest) paths, respectively, and $\Delta\tau_{ij}^{b}(t)$ and $\Delta\tau_{ij}^{w}(t)$ are the additional pheromone increments for them at the t-th iteration, respectively.

Moreover, to prevent excessively high or low pheromone concentration of local paths, which would lead to premature convergence of the algorithm [34], the maximum-minimum ant scheme is employed in this article to keep the pheromone concentration within the range of $[\tau_{\min}, \tau_{\max}]$ by

$$\tau_{ij} = \begin{cases} \tau_{\min}, & \tau_{ij} < \tau_{\min} \\ \tau_{ij}, & \tau_{\min} \leq \tau_{ij} \leq \tau_{\max} \\ \tau_{\max}, & \tau_{ij} > \tau_{\max} \end{cases} \tag{3.8}$$

where $\tau_{\min}$ and $\tau_{\max}$ are the minimum and maximum pheromone concentrations, respectively.

Remark 2. *The performance of the improved ACO algorithm has been enhanced from three aspects: 1) The path initialization method based on the A* algorithm avoids blind searching; 2) The improved heuristic function strengthens the guidance towards the destination; 3) The enhanced pheromone updating method improves the guidance role of excellent paths. These improvements can not only accelerate the convergence speed of the algorithm but also result in better path planning outcomes.*

Based on the aforementioned improved ACO, the global path planning algorithm for mobile robots is summarized in Algorithm 1.

Algorithm 1 Global path planning algorithm

```
Input: Map, start $S$, goal $G$
Output: Optimal global path
Initialize maximum iteration $T$, ant number $N$, start $S$, goal $G$, and factors $\alpha$, $\beta$, $\lambda$, $Q$, $\rho$, $\sigma$, $\tau_{\mathrm{init}}$, $\tau_{\max}$, $\tau_{\min}$
Initialize the pheromone for the grids of the map by Eq (3.5)
for each $t \in [1, T]$ do
    Put all ants into $S$
    for each $k \in [1, N]$ do
        while the $k$-th ant is not in $G$ do
            Choose the next grid by Eqs (3.1) and (3.6)
        end while
        $L_k \leftarrow$ the path length of the $k$-th ant
    end for
    Select the best and worst ants
    for each $k \in [1, N]$ do
        Update the pheromone concentration by Eqs (3.3), (3.4), and (3.8)
    end for
    Update the pheromone concentration of the best and worst paths by Eqs (3.7) and (3.8)
end for
return Optimal global path
```

4. Dynamic path planning based on improved DWA

4.1. Traditional DWA algorithm

DWA is a frequently used algorithm in mobile robot navigation and path planning, aimed at assisting mobile robots in moving safely within unknown environments [35]. Its main principle is to dynamically adjust the mobile robot's speed and direction to choose the optimal motion strategy within a specified time window, thereby avoiding collisions and achieving predefined objectives [36].

In DWA, if the mobile robot moves at a constant velocity within a sufficiently short time interval, it is possible to approximate the mobile robot's trajectory using arcs and straight lines, thereby simplifying the path planning and motion prediction of the mobile robot [16]. Assuming the mobile robot moves in a constant linear velocity over a short time interval Δt, its kinematic model can be expressed as

$$\begin{cases} x(t+\Delta t) = x(t) + v(t)\Delta t \cos\theta(t), \\ y(t+\Delta t) = y(t) + v(t)\Delta t \sin\theta(t), \\ \theta(t+\Delta t) = \theta(t) + w\Delta t, \end{cases} \tag{4.1}$$

where $x(t)$, $y(t)$, and $\theta(t)$ represent the mobile robot's position coordinates and orientation angle at time t, and $v(t)$ and $w(t)$ denote the corresponding linear and angular velocities of the mobile robot.

To choose the optimal motion strategy in the velocity space based on the mobile robot's model and environmental information, the following constraints are usually considered [20]:

- Velocity constraints:

  $$V_r = \Big\{(v, w)|v \in [v_{\min}, v_{\max}], w \in [w_{\min}, w_{\max}]\Big\}$$

  where V_r represents the set of all velocities that the mobile robot can reach under the maximum and minimum constraints of the linear and angular velocities, where $[v_{\min}, v_{\max}]$ and $[w_{\min}, w_{\max}]$ are their limitations, respectively.
- Acceleration and deceleration constraints:

  $$V_a = \Big\{(v, w)|v \in [v_c - \dot{v}_d\Delta t, v_c + \dot{v}_a\Delta t], w \in [w_c - \dot{w}_d\Delta t, w_c + \dot{w}_a\Delta t]\Big\}$$

  where V_a represents the set of all velocities subject to the acceleration and deceleration constraints of the mobile robot due to the limited torque of drive motors, v_c and w_c are current linear and angular velocities, and $\dot{v}_a$ $(\dot{v}_d)$ and $\dot{w}_a$ $(\dot{w}_d)$ indicate the linear and angular acceleration (deceleration) constraints, respectively.
- Braking distance constraints:

  $$V_b = \Big\{(v, w)|v \leq \sqrt{2\delta(v, w)\dot{v}_b}, w \leq \sqrt{2\delta(v, w)\dot{w}_b}\Big\}$$

  where V_b represents the set of all velocities that the robot can stop before colliding with an obstacle in the case of maximum deceleration, and $\delta(v, w)$ represents the distance between the mobile robot's predicted trajectory and the nearest obstacle.

As such, the velocity space of the mobile robot is expressed as

$$V = V_r \cap V_a \cap V_b. \tag{4.2}$$

Based on the velocity space of the mobile robot, several feasible trajectories can be obtained. To identify the optimal trajectory, the traditional DWA typically assesses the trajectories using the following evaluation function [15]:

$$\mathcal{G}(v, w) = \pi(c_1\phi(v, w) + c_2\delta(v, w) + c_3\vartheta(v, w)), \tag{4.3}$$

where $\phi(v, w) = 180^\circ - \theta$ is the azimuth function evaluating the angular deviation θ between the target point and the end of the trajectory corresponding to the current velocity (v, w), $\delta(v, w)$ is the evaluation function indicating the distance of the mobile robot from the obstacle, $\vartheta(v, w)$ is the velocity evaluation function of the trajectory, and c_1–c_3 are respectively the weights assigned to the elements of the trajectory evaluation function, while π stands for the normalization factor.

Remark 3. *Traditional DWA conducts real-time path planning based on locally detected environmental information, but it has the following deficiencies: 1) The robot is prone to making sharp turns at the initial stage when the global path's forward direction deviates significantly from the robot's initial orientation angle; 2) The path planning focuses on real-time obstacle avoidance within the environment based on the guiding role of the target direction, which can easily cause getting stuck in local optima, resulting in the inability to reach the target point; 3) The planned trajectory may not align closely with the global path when DWA is integrated with a global path planning algorithm, leading to unnecessary redundant changes in direction during the robot's movement.*

4.2. *Improved DWA algorithm*

To address the above-mentioned problems, we delete the redundant nodes on the global path generated by the improved ACO, optimize the starting pose of the mobile robot, and refine the evaluation function of the traditional DWA in this article. Ultimately, an improved DWA algorithm incorporating global path information is designed to navigate the mobile robot along the globally planned path while avoiding randomly appearing obstacles, resulting in an optimal path with dynamic autonomous obstacle avoidance capabilities.

4.2.1. Deletion of redundant nodes

The paths planned by the ACO algorithm usually contain many turning points. As such, the mobile robot needs to make several unnecessary posture adjustments to change its direction while tracking the path, leading to a decrease in its operational performance. To ensure the smooth motion of the mobile robot, a scheme for deleting redundant nodes is presented in this article to optimize the obtained global path further.

We assume the global path obtained by the improved ACO algorithm consists of a set of n nodes. Taking the scene depicted in Figure 3 for example, the nodes of the path (the blue dotted line) are numbered as P_1, $P_2, \cdots, P_n$ ($n = 14$ in the figure) from the starting node to the goal node. First, we take P_1 as the current node and connect the nodes P_1 and P_2 to judge whether the segment P_1P_2 intersects an obstacle. If it does not intersect any obstacles, we proceed to link P_1 to the next node until P_1P_j ($j = 3, 4, \cdots, 14$) intersects an obstacle. This indicates that the nodes P_2 to P_{j-2} are candidate redundant nodes that can be removed. Next, we check whether a better path can be obtained from the candidate nodes to the node P_j, and delete the real redundant nodes. Then, we update the current node and repeat the above process starting from the current node until all nodes are checked. It is obvious that the resultant path (the red line in Figure 3) is shorter and has fewer turning points compared to the original path without deleting redundant nodes. The readers can refer to Algorithm 2 for more details on the deletion of redundant nodes.

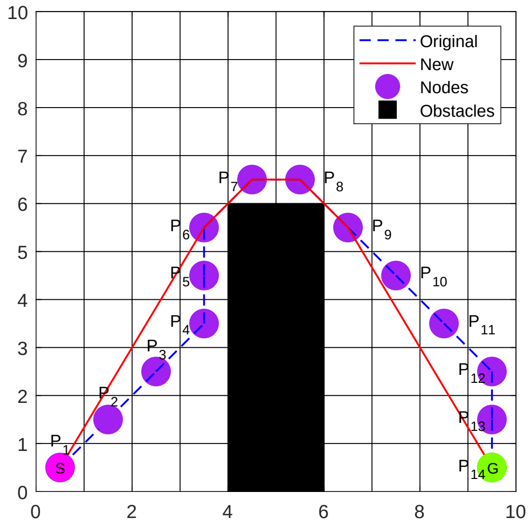


Figure 3. An example for deleting redundant nodes.

Algorithm 2 Dynamic path planning algorithm

```
Input: Map, start $S$, goal $G$, nodes of the global path
Output: Optimal dynamic path
for each grid $i \in [1, n]$ do
    for each grid $j \in [i + 1, n]$ do
        while $P_iP_j$ does not intersect an obstacle do
            $j \leftarrow j + 1$
        end while
        if $\| P_{i+1}P_j \| < \| P_{i+1}P_{j-1} \| + \| P_{j-1}P_j \|$ then
            Retain the effective grid $i$, $G_n \leftarrow i$
        else
            Delete the redundant grid $i$, $G_n \nleftarrow i$
        end if
    end for
end for
Initialize the velocity space $V$, and factors $c_1 - c_5$
Set the initial orientation by Eq (4.4)
while the mobile robot is not in $G$ do
    Sample $m$ velocities from the velocity space $V$
    for each $k \in [1, m]$ do
        Generate the $k$-th simulated trajectory by Eq (4.1)
    end for
    Evaluate the generated $m$ trajectories by Eq (4.7)
    Select the optimal trajectory of $m$ trajectories
    Move the mobile robot along the optimal trajectory
end while
return Optimal dynamic path
```

4.2.2. Setting of the initial orientation

In traditional DWA, the initial orientation angle of a mobile robot typically randomly assigned or set based on prior experience often differs from the forward direction of the global path. When there is a notable angular deviation between the initial orientation angle and the angle to the target point, it would result in unnecessary path searching.

To deal with this problem, in this article, the initial orientation angle of the mobile robot is set based on the direction of the line connecting the robot's starting point to the first sub-goal point on the global path, therefore establishing a clear forward direction and helping avoid additional unnecessary paths. Specifically, this angle is calculated by:

$$\phi = \arctan \frac{y_1 - y_s}{x_1 - x_s}, \tag{4.4}$$

where (x_s, y_s) and (x_1, y_1) are the coordinates of the starting point and the first sub-goal point of the global path nodes, respectively.

4.2.3. Improved evaluation function

In traditional DWA, to ensure the safety of the planned path, the weight of the distance evaluation component in the evaluation function is typically higher. This means that the algorithm tends to prioritize selecting path points that are far away from obstacles, which often results in the mobile robot being unable to reach the target point when there are obstacles around it. In response to this situation and inspired by [15], we introduce a component in the evaluation function representing the distance from the endpoint of the predicted trajectory to the target point, which is expressed as

$$\delta_t(v,w) = \sqrt{(x_e - x_t)^2 + (y_e - y_t)^2}, \tag{4.5}$$

where (x_e, y_e) and (x_t, y_t) are the coordinates of the endpoint of the predicted trajectory and the target point, respectively.

Furthermore, the local path generated by traditional DWA algorithms is sometimes far from the global optimal path. To address this issue, we incorporate a distance evaluation component into the evaluation function to represent the distance from the endpoint of the predicted trajectory to the next node on the global path, which is expressed as

$$\delta_g(v,w) = \sqrt{(x_e - x_g)^2 + (y_e - y_g)^2}, \tag{4.6}$$

where (x_g, y_g) is the coordinate of the next node closest to the endpoint of the predicted trajectory.

Based on the aforementioned improvement schemes, the final evaluation function can be formulated as

$$\mathcal{G}(v,w) = \pi(c_1\phi(v,w) + c_2\kappa(v,w) + c_3\vartheta(v,w) + c_4\delta_t(v,w) + c_5\delta_g(v,w)), \tag{4.7}$$

where c_4 and c_5 are the weights of $\delta_t(v,w)$ and $\delta_g(v,w)$, respectively.

Remark 4. *The improved DWA algorithm reduces the overall path length by removing redundant nodes from the global path and optimizing the initial path direction angle. The enhanced evaluation function includes constraints on the target node and the global path, resulting in path-planning outcomes with a stronger target orientation and approaching a globally optimal path.*

Based on the above improved DWA, the dynamic path planning algorithm for mobile robots is summarized in Algorithm 2, where $\|\cdot\|$ denotes the Euclidean distance of the nodes, G_n is the set of nodes on the global optimal path after deleting redundant nodes, and m is the number of sampling velocities.

5. Simulation results

In this section, several simulation experiments are conducted to confirm the effectiveness and superiority of the proposed algorithms on path planning in global and dynamic environments. In the simulation of path planning algorithms, we consider a two-wheeled differential drive mobile robot.

5.1. Global path planning

To validate the performance of the improved ACO algorithm in the context of global path planning for mobile robots, we will compare it with the traditional ACO algorithm on two maps. To ensure a fair comparison, the same parameters are used for both algorithms in the following manner: $T = 100$ for the maximum iteration, $N = 50$ for the number of ants, $\lambda = 0.3$ for the pheromone diffusion factor, $\alpha = 1$ for the factor of pheromone concentration, $\beta = 5$ for the factor of grid attraction, $\rho = 2.6$ for the factor of grids in the region R, $\sigma = 0.6$ for the weighting parameter of two distances, and $Q = 10$ for the strength of pheromone.

Figure 4 presents the path planning simulation results in a 20 × 20 grid static environment, where the red path is the result of the improved ACO algorithm, and the blue dotted path is the result of traditional ACO algorithm. Figure 5 shows the convergence curves of the improved ACO and traditional ACO algorithms, respectively. Table 1 provides a comparison of the results obtained by the two algorithms. From the above results, it can be observed that, compared to the traditional ACO algorithm, the improved ACO algorithm in this article has reduced the path length by 16.99%, the number of convergence iterations by 80.95%, the number of path turning points by 52.94%, and the runtime by 52.93%.

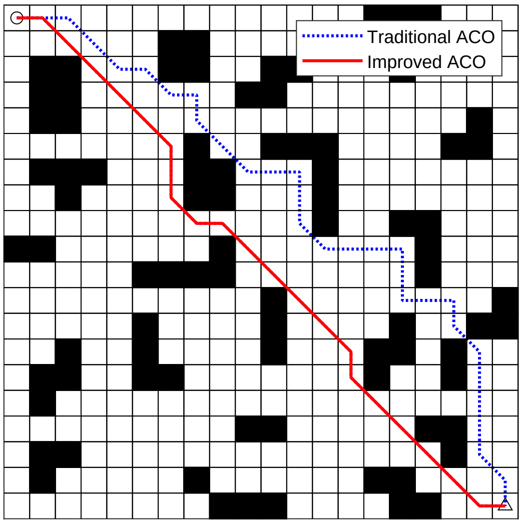


Figure 4. Global path planning on a 20 × 20 map.

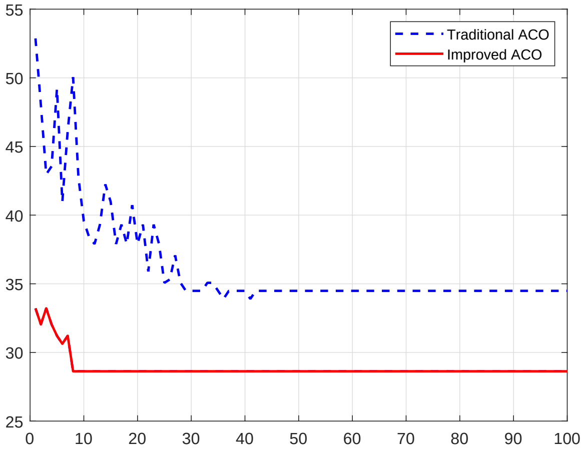


Figure 5. Convergence curves for a 20 × 20 map.

Table 1. Comparison of global path on a 20 × 20 map.

Algorithm	Path length	Convergence iteration	Turning points	Runtime (s)
Traditional ACO	34.48	42	17	10.22
Improved ACO	28.62	8	8	4.81

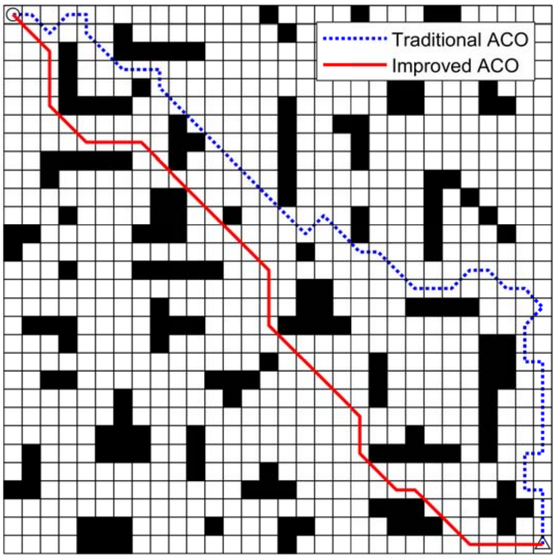


Figure 6. Global path planning on a 30 × 30 map.

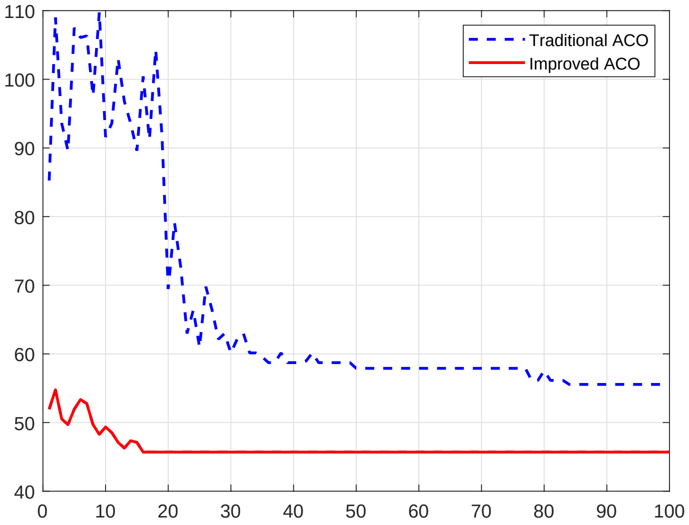


Figure 7. Convergence curves for a 30 × 30 map.

Table 2. Comparison of global path on a 30 × 30 map.

Algorithm	Path length	Convergence iteration	Turning points	Runtime (s)
Traditional ACO	55.55	84	26	29.47
Improved ACO	45.69	16	11	25.38

Furthermore, we conducted comparative experiments in a 30 × 30 grid static environment. From the path planning results and the algorithm convergence curves in Figures 6 and 7, it is evident that the improved algorithm continues to yield better results in this map environment. By examining the comparative data in Table 2, it can be seen that the improved algorithm has decreased the path length by 17.75%, the number of convergence iterations by 80.95%, the number of path turning points by 57.69%, and the runtime by 13.87%, when compared to the traditional ACO algorithm.

The above simulation results demonstrate that the improved ACO algorithm in this study is capable of achieving global path planning results with shorter lengths and fewer turning points. Moreover, it shows enhanced search efficiency and convergence speed compared to the traditional ACO algorithm, which can further highlight the effectiveness and superiority of the improved algorithm.

5.2. Dynamic path planning

In the next step, the dynamic path planning is carried out based on the results obtained from global path planning. To validate the performance of the proposed path planning algorithm for mobile robots in dynamic environments, several simulation experiments are conducted in two grid map environments with different levels of complexity. In the grid map, black grids represent static obstacles, gray squares are newly added obstacles after the global path planning, and the purple square denotes the mobile obstacle moving at 0.4 m/s. The parameter settings for the simulation experiment are as follows: $v_{\max} = 1.5$ m/s and $v_{\min} = 0$ m/s for linear velocity constraints, $w_{\max} = 0.35$ rad/s and $w_{\min} = -0.35$ rad/s for angular velocity constraints, $\dot{v}_a = 0.2$ m/s^2 and $\dot{v}_d = 0$ m/s^2 for linear acceleration and deceleration constraints, $\dot{w}_a = 0.9$ rad/s^2 and $\dot{w}_d = -0.9$ rad/s^2 for angular acceleration and deceleration constraints, the time window for predicting trajectories is 3 seconds, and the factors c_1–c_5 are set as 0.15, 0.15, 0.2, 0.2, and 0.3, respectively.

Figures 8 and 9 present the path planning results in a 20 × 20 grid dynamic environment using the traditional DWA algorithm and the improved DWA algorithm, where the purple moving obstacle travels along the black dotted path as shown in the figures. In Figure 8, the mobile robot has to make additional posture adjustments because its initial orientation is inconsistent with the global path direction, causing the path planning result to deviate significantly from the global path. In contrast, as seen in Figure 9, the proposed algorithm in this article optimizes the robot's initial pose at the starting position, ensuring that the initial motion direction of the mobile robot aligns with the global path direction, thus avoiding unnecessary large turning movements. Furthermore, during the dynamic obstacle avoidance movement, the traditional algorithm tends to deviate significantly from the global optimal path, leading to an increase in the length of the mobile robot's trajectory. In contrast, the algorithm proposed in this article can yield path planning results that are closer to the global optimal path by enhancing the trajectory evaluation function. This result, which is closer to the global path, can also be reflected in the navigation around newly added obstacles as shown in the figures.

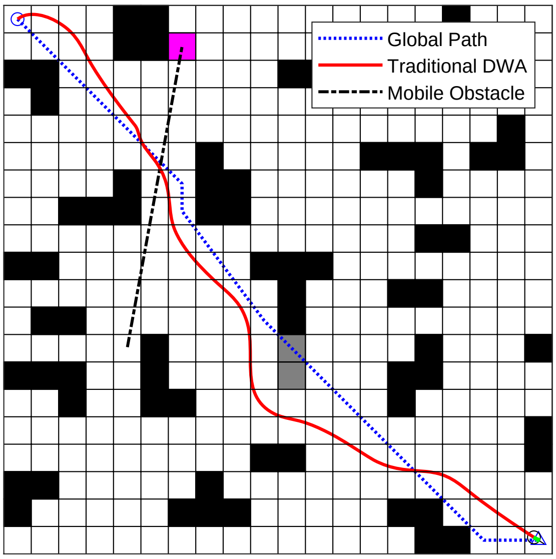


Figure 8. Dynamic path planning on a 20 × 20 map using traditional DWA.

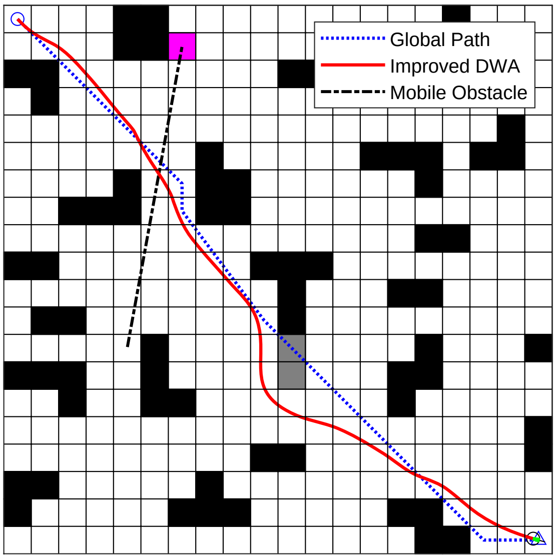


Figure 9. Dynamic path planning on a 20 × 20 map using improved DWA.

In Figures 10 and 11, two obstacle avoidance scenes using improved DWA are depicted. The blue dotted circle represents the robot's detection range, and the green trajectory denotes the feasible paths generated by DWA. In Figure 10, a moving obstacle is within the robot's detection range, causing the feasible local planning paths to shift to the right. In Figure 11, the moving obstacle has moved out of the robot's detection range, allowing the feasible local planning paths to return to the global path. It can be observed that a mobile robot can plan an appropriate obstacle avoidance path based on the situation of obstacles within the detection range and eventually return to the global optimal path.

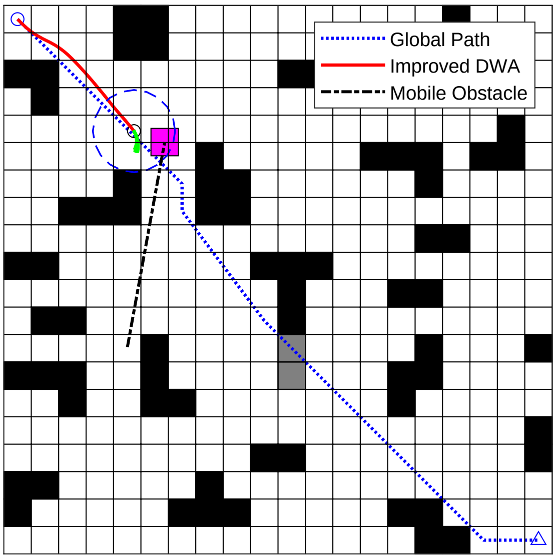


Figure 10. Scene 1 for avoiding moving obstacles using improved DWA.

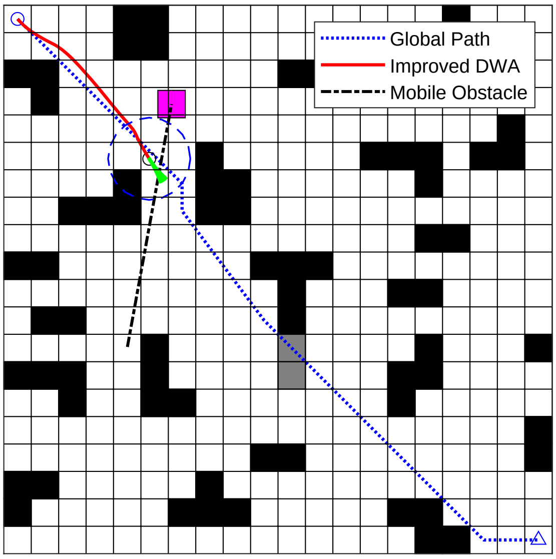


Figure 11. Scene 2 for avoiding moving obstacles using improved DWA.

Table 3 provides a comparison of the results of the aforementioned simulation experiments. It can be observed that, in comparison with traditional methods, the proposed algorithm can reduce path length by 7.97% and decrease runtime by 9.23%, which further confirms the superiority of the algorithm presented in this article. Furthermore, we conduct simulation comparative experiments in a 30 × 30 grid dynamic environment, and the results are shown in Figures 12 and 13. It can be observed that in a larger and more complex environment, the path planning result obtained by the proposed algorithm proposed in this article is closer to the globally optimal path. Table 4 provides a comparison of the simulation experiment results, and it is obvious that the algorithm presented in this article reduces both path length and runtime significantly.

The above simulation results demonstrate that, by deleting redundant nodes, optimizing the initial pose, and improving the trajectory evaluation function, the improved algorithm proposed in this article can successfully obtain collision-free optimal paths in dynamic environment path planning. Compared to traditional algorithms, the obtained paths are shorter, with lower time costs, and the motion trajectory of the mobile robot is smoother.

Table 3. Comparison of dynamic path on a 20 × 20 map.

Algorithm	Path length	Runtime (s)
Traditional DWA	35.54	185.61
Improved DWA	32.71	168.47

Table 4. Comparison of dynamic path on a 30 × 30 map.

Algorithm	Path length	Runtime (s)
Traditional DWA	55.25	262.83
Improved DWA	47.83	192.41

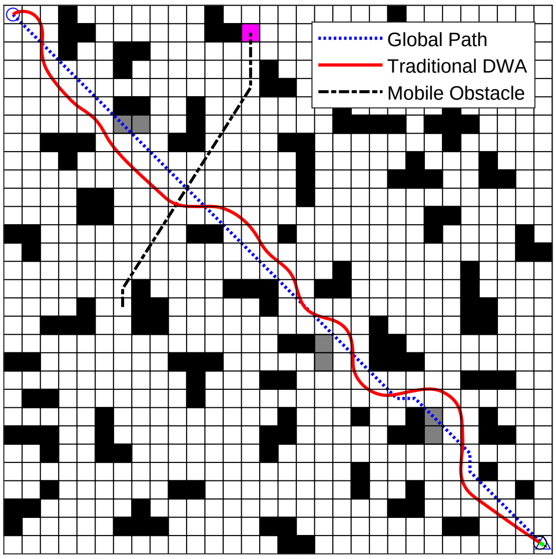


Figure 12. Dynamic path planning on a 30 × 30 map using traditional DWA.

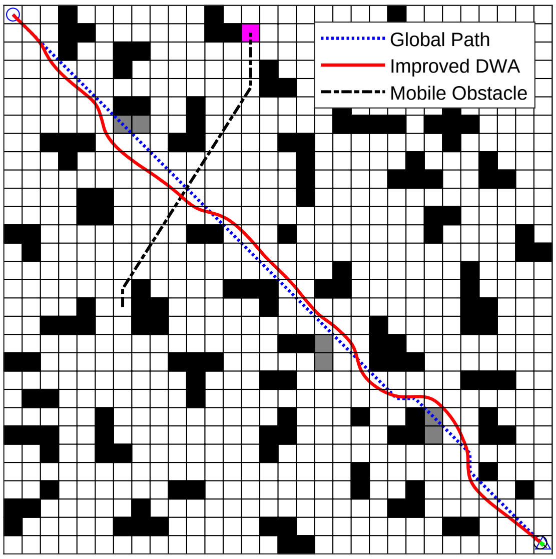


Figure 13. Dynamic path planning on a 30 × 30 map using improved DWA.

6. Conclusions

In this article, a new strategy integrating the improved ACO and DWA algorithms has been proposed for mobile robot path planning in dynamic environments. Through improvements in the initialization of pheromones, heuristic function, and updating of pheromones, the improved ACO can obtain a globally optimal path more efficiently. Based on the globally optimal path, the improved DWA, which is enhanced by deleting the redundant nodes, optimizing the initial orientation, and modifying the evaluation function, has been utilized to produce an optimal path in dynamic environments. The simulation experiments show that the improved algorithms can obtain superior results for path-planning tasks in dynamic environments.

In recent years, researchers have conducted numerous outstanding works in the field of path planning for mobile robots. In future work, we will further draw upon the excellent achievements in the field of intelligent optimization to improve the proposed path planning algorithm in practical applications within dynamic environments [37–40], and deploy it in more complex systems and environments, such as multi-robot systems [41,42], networked robot systems [43,44], and so on.

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 was supported in part by the Natural Science Foundation of Shandong Province of China (ZR2023MF067) and the National Natural Science Foundation of China (62273211, 62273214).

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. S. Zhu, W. Zhu, X. Zhang, T. Cao, Path planning of lunar robot based on dynamic adaptive ant colony algorithm and obstacle avoidance, *Int. J. Adv. Rob. Syst.*, **17** (2020). http://doi.org/10.1177/1729881419898979

2. E. Fernandes, P. Costa, J. Lima, G. Veiga, Towards an orientation enhanced astar algorithm for robotic navigation, in *2015 IEEE International Conference on Industrial Technology (ICIT)*, (2015), 3320–3325. https://doi.org/10.1109/ICIT.2015.7125590

3. 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

4. G. Che, L. Liu, Z. Yu, An improved ant colony optimization algorithm based on particle swarm optimization algorithm for path planning of autonomous underwater vehicle, *J. Ambient Intell. Hum. Comput.*, **11** (2020), 3349–3354. https://doi.org/10.1007/s12652-019-01531-8

5. A. Yazici, G. Kirlik, O. Parlaktuna, A. Sipahioglu, A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints, *IEEE Trans. Cybern.*, **44** (2014), 305–314. https://doi.org/10.1109/TCYB.2013.2253605

6. L. Liu, J. Lin, J. Yao, D. He, J. Zheng, J. Huang, et al., Path planning for smart car based on dijkstra algorithm and dynamic window approach, *Wireless Commun. Mobile Comput.*, **2021** (2021), 8881684. https://doi.org/10.1155/2021/8881684

7. F. Duchon, A. Babinec, M. Kajan, P. Beno, M. Florek, T. Fico, et al., Path planning with modified a star algorithm for a mobile robot, *Procedia Eng.*, **96** (2014), 59–69. https://doi.org/10.1016/j.proeng.2014.12.098

8. Z. Wu, J. Dai, B. Jiang, H. R. Karimi, Robot path planning based on artificial potential field with deterministic annealing, *ISA Trans.*, **138** (2023), 74–87. https://doi.org/10.1016/j.isatra.2023.02.018

9. N. Wang, H. Xu, C. Li, J. Yin, Hierarchical path planning of unmanned surface vehicle: A fuzzy artificial potential field approach, *Int. J. Fuzzy Syst.*, **23** (2021), 1797–1808. https://doi.org/10.1007/s40815-020-00912-y

10. E. Li, K. Qi, Ant colony algorithm for path planning based on grid feature point extraction, *J. Shanghai Jiaotong Univ. (Sci.)*, **28** (2023), 86–99. https://doi.org/10.1007/s12204-023-2572-4

11. N. Zeng, H. Zhang, Y. Chen, B. Chen, Y. Liu, Path planning for intelligent robot based on switching local evolutionary PSO algorithm, *Assem. Autom.*, **36** (2016), 120–126. https://doi.org/10.1108/AA-10-2015-079

12. J. Ballesteros, C. Urdiales, A. B. M. Velasco, G. Ramos-Jimenez, A biomimetical dynamic window approach to navigation for collaborative control, *IEEE Trans. Hum. Mach. Syst.*, **47** (2017), 1123–1133. https://doi.org/10.1109/THMS.2017.2700633

13. J. Kim, G. Yang, Improvement of dynamic window approach using reinforcement learning in dynamic environments, *Int. J. Control Autom. Syst.*, **20** (2022), 2983–2992. https://doi.org/10.1007/s12555-021-0462-9

14. W. Yang, P. Wu, X. Zhou, H. Lv, X. Liu, G. Zhang, et al., Improved artificial potential field and dynamic window method for amphibious robot fish path planning, *Appl. Sci.*, **11** (2021), 2114. https://doi.org/10.3390/app11052114

15. B. Wu, X. Chi, C. Zhao, W. Zhang, Y. Lu, D. Jiang, Dynamic path planning for forklift AGV based on smoothing A* and improved DWA hybrid algorithm, *Sensors*, **22** (2022), 7079. https://doi.org/10.3390/s22187079

16. L. Chang, L. Shan, Y. Dai, Z. Qi, Multi-robot formation control in unknown environment based on improved DWA (in Chinese), *Control Decis.*, **37** (2021), 2524–2534. http://doi.org/10.13195/j.kzyjc.2020.1817

17. M. Si, X. Zhou, Y. Zhang, Improvement of dynamic window approach in dynamic obstacle environment, *J. Phys. Conf. Ser.*, **2477** (2023), 012059. https://doi.org/10.1088/1742-6596/2477/1/012059

18. Y. Peng, Z. Huang, S. Li, Research on automatic obstacle avoidance navigation of mobile robot based on dynamic window method (in Chinese), *Autom. Instrum.*, **41** (2020), 26–29+33. https://doi.org/10.16086/j.cnki.issn1000-0380.2020020004

19. Y. Wang, Y. Tian, X. Li, L. Li, Self-adaptive dynamic window approach in dense obstacles (in Chinese), *Control Decis.*, **34** (2018), 927–936. https://doi.org/10.13195/j.kzyjc.2017.1497

20. H. Zhang, M. Li, Rapid path planning algorithm for mobile robot in dynamic environment, *Adv. Mech. Eng.*, **9** (2017). https://doi.org/10.1177/1687814017747400

21. F. Wang, T. Li, J. Liu, H. Zhao, Research on autonomous path planning and obstacle avoidance of construction robot based on BIM (in Chinese), *Comput. Eng. Appl.*, **56** (2020), 224–230.

22. C. Lao, P. Li, Y. Feng, Path planning of greenhouse robot based on fusion of improved A* algorithm and dynamic window approach (in Chinese), *Trans. Chin. Soc. Agric. Mach.*, **52** (2021), 14–22. https://doi.org/10.6041/j.issn.1000-1298.2021.01.002

23. Y. Li, J. Zhao, Z. Chen, G. Xiong, S. Liu, A robot path planning method based on improved genetic algorithm and improved dynamic window approach, *Sustainability*, **15** (2023), 4656. https://doi.org/10.3390/su15054656

24. L. Yang, L. Fu, P. Li, J. Mao, N. Guo, An effective dynamic path planning approach for mobile robots based on ant colony fusion dynamic windows, *Machines*, **10** (2022), 50. https://doi.org/10.3390/machines10010050

25. Y. Zhang, H. Quan, J. Wen, Mobile robot path planning based on the wolf ant colony hybrid algorithm (in Chinese), *J. Huazhong Univ. Sci. Technol. (Nat. Sci.)*, **48** (2020), 127–132. https://doi.org/10.13245/j.hust.200123

26. W. Wang, J. Zhao, Z. Li, J. Huang, Smooth path planning of mobile robot based on improved ant colony algorithm, *J. Rob.*, **2021** (2021), 4109821. https://doi.org/10.1155/2021/4109821

27. W. Hou, Z. Xiong, C. Wang, H. Chen, Enhanced ant colony algorithm with communication mechanism for mobile robot path planning, *Rob. Auton. Syst.*, **148** (2022), 103949. https://doi.org/10.1016/j.robot.2021.103949

28. W. Li, L. Xia, Y. Huang, M. Soroosh, An ant colony optimization algorithm with adaptive greedy strategy to optimize path problems, *J. Ambient Intell. Hum. Comput.*, **13** (2021), 1557–1571. https://doi.org/10.1007/s12652-021-03120-0

29. L. Xu, M. Cao, B. Song, A new approach to smooth path planning of mobile robot based on quartic Bezier transition curve and improved PSO algorithm, *Neurocomputing*, **473** (2022), 98–106. https://doi.org/10.1016/j.neucom.2021.12.016

30. X. Xiong, H. Min, Y. Yu, P. Wang, Application improvement of A* algorithm in intelligent vehicle trajectory planning, *Math. Biosci. Eng.*, **18** (2020), 1–21. https://doi.org/10.3934/mbe.2021001

31. X. Dai, S. Long, Z. Zhang, D. Gong, Mobile robot path planning based on ant colony algorithm with A* heuristic method, *Front. Neurorobot.*, **13** (2019), 1–9. https://doi.org/10.3389/fnbot.2019.00015

32. 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

33. 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

34. B. Song, H. Miao, L. Xu, Path planning for coal mine robot via improved ant colony optimization algorithm, *Syst. Sci. Control Eng.*, **9** (2021), 283–289. https://doi.org/10.1080/21642583.2021.1901158

35. D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, *IEEE Rob. Autom. Mag.*, **4** (1997), 23–33. https://doi.org/10.1109/100.580977

36. X. Lai, D. Wu, D. Wu, J. Li, H. Yu, Enhanced DWA algorithm for local path planning of mobile robot, *Ind. Rob.*, **50** (2023), 186–194. https://doi.org/10.1108/IR-05-2022-0130

37. Z. Mei, T. Zhao, X. Xie, Hierarchical fuzzy regression tree: A new gradient boosting approach to design a TSK fuzzy model, *Inf. Sci.*, **652** (2024), 119740. https://doi.org/10.1016/j.ins.2023.119740

38. Z. Wu, H. R. Karimi, C. Dang, An approximation algorithm for graph partitioning via deterministic annealing neural network, *Neural Networks*, **117** (2019), 191–200. https://doi.org/10.1016/j.neunet.2019.05.010

39. H. Chen, C. Li, M. Mafarja, A. A. Heidari, Y. Chen, Z. Cai, Slime mould algorithm: A comprehensive review of recent variants and applications, *Int. J. Syst. Sci.*, **54** (2023), 204–235. https://doi.org/10.1080/00207721.2022.2153635

40. Z. Yu, P. Duan, L. Meng, Y. Han, F. Ye, Multi-objective path planning for mobile robot with an improved artificial bee colony algorithm, *Math. Biosci. Eng.*, **20** (2023), 2501–2529. https://doi.org/10.3934/mbe.2023117

41. J. Wang, Y. Zhuang, Y. Liu, FSS-Net: A fast search structure for 3D point clouds in deep learning, *Int. J. Network Dyn. Intell.*, **2** (2023), 100005. https://doi.org/10.53941/ijndi.2023.100005

42. Y. Wang, W. Liu, C. Wang, F. Fadzil, S. Lauria, X. Liu, A novel multi-objective optimization approach with flexible operation planning strategy for truck scheduling, *Int. J. Network Dyn. Intell.*, **2** (2023), 100002. https://doi.org/10.53941/ijndi.2023.100002

43. L. Zou, Z. Wang, B. Shen, H. Dong, G. Lu, Encrypted finite-horizon energy-to-peak state estimation for time-varying systems under eavesdropping attacks: Tackling secrecy capacity, *IEEE/CAA J. Autom. Sin.*, **10** (2023), 985–996. https://doi.org/10.1109/JAS.2023.123393

44. L. Zou, Z. Wang, B. Shen, H. Dong, Moving horizon estimation over relay channels: Dealing with packet losses, *Automatica*, **155** (2023), 111079. https://doi.org/10.1016/j.automatica.2023.111079