MDPI

Article

NMS-EACO: A Novel Multi-Strategy ACO for Mobile Robot Path Planning

Chao Zhang [1], Jing Ma [2], Xin Wang [1,*], Jianwei Xu [1] and Chuanchen Guo [3]

[1] Institute of Automation, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250013, China, zc2011@qlu.edu.cn, 10431230622@stu.qlu.edu.cn
[2] School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, 1010, New Zealand, jing.ma@aut.ac.nz
[3] Testing Center, Tiezheng Testing Technology Co., Ltd., Jinan 250014, China, zz8454984@gmail.com
* Correspondence: xinwang@sdas.org

Abstract: Ant Colony Optimization (ACO) has been widely used in engineering implementation due to its simplicity and effectiveness. However, it often faces challenges such as slow convergence, susceptibility to local optima, and generating paths with excessive turning points. To address these limitations, this paper introduces NMS-EACO, a novel multi-strategy enhanced ant colony optimization algorithm for mobile robot path planning under nonholonomic constraints. NMS-EACO integrates five key strategies: an A*-guided heuristic function, an adaptive enhanced pheromone update rule, a state transition probability under nonholonomic constraints, a smoothing factor embedded in the state transition probability, and a global path smoothing technique. Comprehensive simulation experiments are conducted across three distinct map types, with comparisons made against four existing algorithms through extensive trials. Results demonstrate that NMS-EACO significantly improves convergence speed, enhances global search capability, and reduces path irregularities. These results validate the robustness and efficiency of the proposed multi-strategy method for nonholonomic mobile robot navigation

Keywords: Ant colony optimization; ACO; Path planning; Mobile robots; Nonholonomic constraints

1. Introduction

Mobile robots, as an emerging form of transport vehicles, are widely utilized across multiple applications due to their high flexibility and efficient transportation capabilities [1,2]. However, the diversity of their use cases, the complexity of spatial layout, and the demand for operational efficiency present significant challenges to achieving fully autonomous navigation [30]. Among the core techniques, path planning plays a critical role in determining both the precision and efficiency of mobile robot operations [3]. The effectiveness of the planning algorithm directly impacts the robot's ability of tasks accuracy and autonomous functionality [4,5]. Therefore, designing an optimal path for mobile robots remains a fundamental problem in this domain [6].

Since the last century, extensive research have been conducted on robot path planning, resulting in numerous well-designed algorithms [31], such as the A* algorithm [32]. In recent years, the focus of research has shifted toward more sophisticated approaches tailored for complex and dynamic environments [33]. For instance, [7] introduces a sine resistance network-based method for autonomous electric vehicles in dynamic settings, while [8] develops a self-supervised cost of transport estimation method to assist multi-modal robots navigation. Similarly, [9] proposes a kinematically constrained batch informed

Received:
Revised:
Accepted:
Published:

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Journal Not Specified* **2025**, *1*, 0. https://doi.org/

trees (BIT*) algorithm utilizing variable density sampling to enhance efficiency and accuracy. [10] presents edge computing to optimize vehicle path planning by minimizing delay and energy consumption. Although the above methods demonstrate strong theoretical performance, their practical deployment remains limitation due to complexity and implementation challenges.

Ant Colony Optimization (ACO), a nature-inspired algorithm based on the foraging behavior of ants [11], has shown promise in addressing practical path planning problems, such as avoiding long paths [12]. Although ACO has the advantages of high efficiency and robustness [13], it suffers from the issues of slow convergence speed, susceptibility to local optima, and excessive path turning points. Recent efforts to enhance ACO have tackled these issues from multiple perspectives. For example, [14] introduces a potential field enhanced ACO that integrates a repulsive field function and adjustment factors to improve pheromone updates, effectively reducing path length. In [15], a local pheromone update mechanism is used to lower pheromone concentration at visited nodes, encouraging population diversity, however, its fixed update mechanism limits adaptability. [16] uses multiple swarm collaboration strategies to effectively improve global optimization ability and convergence efficiency by strategically distributing ants in starting and goal regions. Additionally, [17] introduces a greedy initialization strategy to guide early pheromone distribution and speed up convergence. A negative feedback balancing factor is presented in [18] along with extended ants independent of the main pheromone trail, helping to escape local optima in later stages.

Although these approaches have achieved substantial improvements, some issues of the algorithm remain partially unsolved, for instance, slow convergence speed, excessive path turning points. To address these limitations, this paper proposes a novel multi-strategy enhanced ACO under nonholonomic constraints(NMS-EACO). The proposed NMS-EACO incorporates five mechanisms: A*-guided heuristic function, adaptive enhanced pheromone update rule, state transition probability under nonholonomic constraints, smooth factor for optimizing state transition probability, global path smoothing strategy. To validate the effectiveness of NMS-EACO, extensive simulation experiments—comprising hundreds of test cases across three types of maps—are conducted. The results demonstrate significant improvements in convergence speed, global search capability, and path smoothness compared to existing methods.

The structure of this paper is organized as follows: Section 2 presents the ACO's theoretical basis and mathematical model. Section 3 presents five innovative mechanisms. Section 4 describes the simulation experiments, and analyzes experimental results. Section 5 gives the conclusions of this article.

2. Mathematical Model and Working Environment

ACO is inspired by the natural behavior of ants, which communicate by releasing pheromones and collaborate to discover the shortest path while foraging for food [19]. In this paper, the mobile robot is abstracted as an ant, and its position within the grid environment is represented by coordinates, effectively simulating the robot's real-world operating conditions.

The ant determines its next target point based on a combination of pheromone concentration and heuristic information. The probability of an ant transitioning from position i to position j is defined by the state transition formulas presented in Equations (1) and (2).

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

$$\begin{cases} \eta_{ij}(t) = \dfrac{1}{d_{ij}} \\ d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \end{cases} \tag{2}$$

where $p_{ij}^k(t)$ denotes the probability that the k-th ant moves from its current position i to position j at time t. The condition $j \in \text{allowed}_k$ indicates that node j has not yet been visited by ant k, making it a valid candidate for the next step. $\tau_{ij}(t)$ represents the pheromone level on the path between nodes (i, j) at time t, while $\eta_{ij}(t)$ is the heuristic value that guides the k-th ant in choosing the path from i to j. The parameter α is the heuristic factor of pheromone, indicating the influence of pheromone levels on path selection. β is the factor of heuristic function, reflecting the weight of heuristic information in the decision-making process. Lastly, d_{ij} denotes the Euclidean distance between nodes i and j.

During the foraging process, ants release pheromones along the paths they travel, which serves to guide other ants in selecting their routes [20]. Meanwhile, these pheromones gradually evaporate over time. After all k ants have completed an iteration, the global pheromone levels are updated according to Equations (3) and (4).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{K} \Delta\tau_{ij}^k(t) \tag{3}$$

$$\Delta\tau_{ij}^k(t) = \frac{Q}{L_k} \tag{4}$$

where $\boldsymbol{\rho}$ is the global pheromone evaporation coefficient, ranging between 0 and 1, where a higher value indicates faster pheromone evaporation. $\Delta\tau_{ij}^k$ indicates the amount of pheromone released by ant k on the path (i, j) during the current iteration. Q is the pheromone enhancement coefficient and L_k represents the total length of the path traveled by ant k.

The working environment of a mobile robot is often complex and abstract, making it essential to model the scene accurately. Common methods for environmental map modeling include the visual graph method, unit decomposition method, and grid method [21]. Among these, the grid method is the most widely used due to its intuitive structure, ease of understanding, and straightforward implementation. In this study, a two-dimensional grid map is constructed using the grid method. As illustrated in Figure 1, $\boldsymbol{S}$ denotes the starting point, $\boldsymbol{E}$ indicates the target endpoint, white cells represent free space, black cells denote obstacles, and each grid cell corresponds to a fixed coordinate.

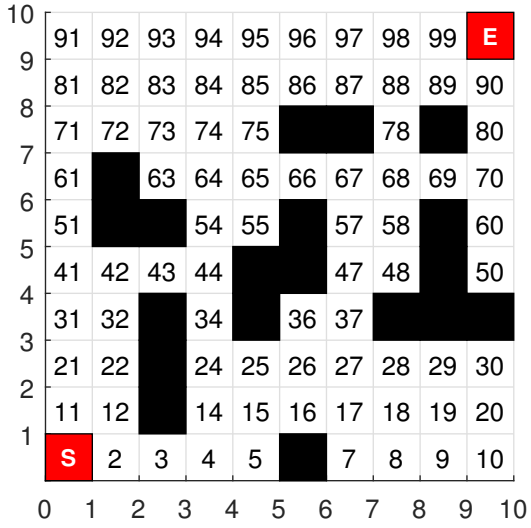


Figure 1. Grid-based Working Scene

3. Multi-Strategy Enhanced ACO with Nonholonomic Constraints (NMS-EACO)

3.1. A-Guided Heuristic Function*

ACO primarily relies on the local distance information between the ant's current node and its neighboring nodes to build the transition probability model during path optimization, which shows significant limitations [22]. Without a global awareness of the solution space, the algorithm is tend to getting trapped in local optima during iterations, leading to frequent path oscillations and an increased number of inflection points. Additionally, the pheromone update rule places too much emphasis on local paths, weakening the algorithm's ability to explore unknown regions and causing the search process to become blind, thereby significantly reducing convergence speed.

In contrast, the A* algorithm employs a heuristic function to estimate the cost from the current node to the target, effectively reducing unnecessary node expansion and improving search efficiency [23,24]. Building on this, the paper incorporates the distance between the ant's selected position and the target point, introducing an adaptive adjustment factor $\boldsymbol{u}$ and an adaptive distance weight coefficient ε. The improved heuristic functions are defined in Equations (5) and (6).

$$\eta_{ij}(t) = \frac{u}{e^{\varepsilon d_{ij} + (1-\varepsilon) d_{je}}} \tag{5}$$

$$u = exp\left(-\left(\frac{N}{N_{max}}\right)^2\right) \tag{6}$$

where d_{ij} represents the distance between the ant's current position i and the candidate position j, while d_{je} denotes the distance from position j to the target point e. The parameter u is the adaptive adjustment factor, N refers to the current iteration number, and N_{max} is the maximum number of iterations allowed.

As illustrated in Equation (6), the adaptive adjustment u decreases progressively as the iteration count N increases. Consequently,the impact of the heuristic function on the ant's state transition probability shifts from being nearly 1 at the start to almost 0 in the later stages of the algorithm's execution. This strategy enhances the algorithm's convergence speed while preserving its ability to perform global search.

3.2. Adaptive Improved Pheromone Update Strategy

In practical applications, using the same pheromone update method for all ants can cause the algorithm to become trapped in local optima and slow down convergence. To address these issues, this paper proposes an adaptive enhanced elite ant strategy.

Traditionally, the elite ant strategy designates the ant with the shortest path in each iteration as the elite ant. Only the pheromone on the elite ant's path is updated, while pheromone contributions from ants stuck in deadlocks are discarded to prevent indiscriminate global pheromone updates. However, when multiple ants traverse the same path, the pheromone accumulation can exceed its evaporation rate. As a result, even if elite ants discover a better path, the higher pheromone concentration on existing paths may prevent the elite ant strategy from effectively improving the solution.

To overcome this, an adaptive enhancement factor $\boldsymbol{q}$ is introduced to amplify the pheromone deposited by elite ants, as detailed in Equation (7).

$$q = \begin{cases} e^{\frac{N-1}{N_{max}}} - 1, L_b < L \\ 0, otherwise \end{cases} \tag{7}$$

where L_b represents the optimal path length found in the current iteration, while L denotes the best historical path length up to the current time. N_{max} is the maximum allowed number of iterations, and $\mathbf{N}$ indicates the current iteration count. The adaptive enhancement factor $\boldsymbol{q}$ increases continuously as the number of interations grows.

In addition to incorporating the adaptive enhanced elite ant strategy in the pheromone update rule, this paper also considers the number of turns along the path. To capture this, the turn count Z_K is introduced to record the number of turns encountered during path planning.

The improved pheromone update rule is expressed in Equations (8), (9) and (10).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho \sum_{k=1}^{K} \Delta\tau_{ij}^{k}(t) + q\Delta\tau_{ij}^{b}(t) \tag{8}$$

$$\Delta\tau_{ij}^{k}(t) = \frac{Q}{L_K + YZ_K} \tag{9}$$

$$\Delta\tau_{ij}^{b} = \frac{Q}{L_{min}} \tag{10}$$

where $\Delta\tau_{ij}^{k}$ denotes the pheromones concentration on the path from grid i to j at time t. The parameter $\boldsymbol{\rho}$ represents the global pheromone evaporation coefficient. $\Delta\tau_{ij}^{k}$ is the concentration of pheromones released by ant k after traversing path(i, j) in the current iteration. Q is the pheromone enhancement coefficient, and L_k is the length of the path traveled by ant k in the current iteration. Z_K records the number of turns along path (i, j), while γ is the path smoothness coefficient. $\Delta\tau_{ij}^{b}(t)$ is the concentration of pheromones released by the elite ant b on path(i, j) during the current iteration. L_{min} is the optimal path length in the current iteration, which is shorter than the previous best path length.

3.3. State Transition Probability under Nonholonomic Constraints

In ACO-based path planning, the mobile robot is often simplified as a particle [25], neglecting the nonholonomic constraints of the robot's chassis. This simplification makes it challenging to align the planned path with the actual movement capabilities of the robot, limiting the algorithm's practical applicability [26,27]. The Ackermann steering model is a common chassis mobile robots, is shown in Figure 2, where λrepresents the maximum steering angle of the front wheels, ***WT*** is the wheel track, and ***WB*** is the wheelbase. Due

to these constraints, the mobile robot can only move in directions aligned with its wheels, such as forward, left-front, and front-right. Incorporating this nonholonomic constraint into the ACO framework is described by Equations (11) and (12).

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}[\mu_{ij}(t)]^{\nu}}{\sum_{j \in allowed_k}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}[\mu_{ij}(t)]^{\nu}}, j \in allowed_k \\ 0, \quad otherwise \end{cases} \tag{11}$$

$$\mu_{ij}(t) = \begin{cases} e^{1-\left|\frac{\theta_{ij}(t)}{\pi}\right|}, \theta_{ij}(t) < |\frac{\pi}{2}| \\ 0, \quad \theta_{ij}(t) > |\frac{\pi}{2}| \end{cases} \tag{12}$$

Where: $\mu_{ij}(t)$ is the turning factor, $\theta_{ij}(t)$ is the direction angle of the ant from the current node i to the next node j. For example, when the ant moves straight ahead, $\theta_{ij}(t)$ equals 0. If $\theta_{ij}(t)$ is less than $|\frac{\pi}{2}|$, the turning factor $\mu_{ij}(t)$ decreases as $\theta_{ij}(t)$ increases, encouraging the ant to make smaller turns. when $\theta_{ij}(t)$ exceeds $|\frac{\pi}{2}|$, $\mu_{ij}(t)$ is set to 0 to prevent ant from moving in the opposite direction, thereby better reflecting the nonholonomic constraints of the mobile robot. The parameter ν is the weighting coefficient for the angle factor.

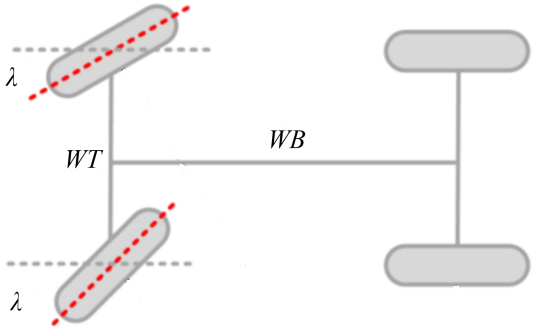


Figure 2. Ackermann Steering Model

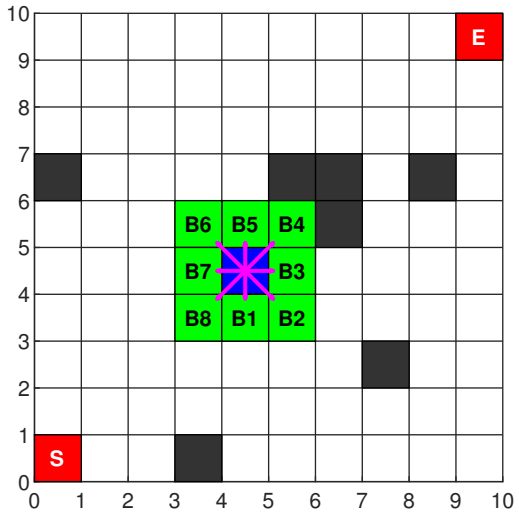


Figure 3. The Selectable Nodes of the Current Node without Nonholonomic Constraints

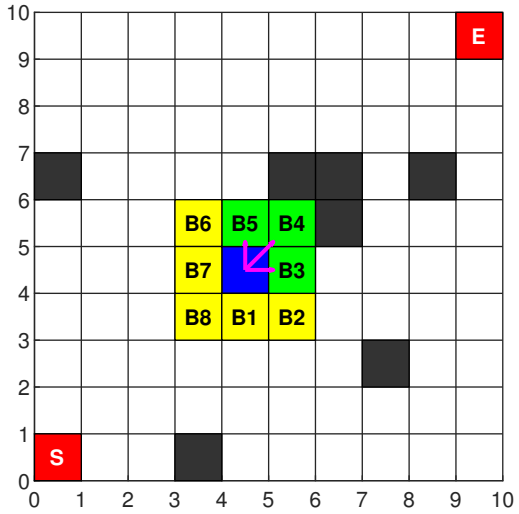


Figure 4. The Selectable Nodes of the Current Node with Nonholonomic constraints.

As shown in Figure 3, when nonholonomic constraints are not considered, the ACO may select any of the eight surrounding nodes (B1 to B8) as the next target point. However, after integrating nonholonomic constraints, as shown in Figure 4, the ants prioritize nodes B3, B4, or B5 as potential target points. This adjustment aligns with the motion limitations of mobile robots imposed by nonholonomic constraints and enhances the efficiency of the path planning process.

3.4. Smooth Factor for Optimizing State Transition Probability

In practical operation, a mobile robot's path should not only be short in distance but also consist of more straight segments with fewer turns. To achieve smoother paths, this paper introduces a smoothing factor into the state transition probability equation. The enhanced state transition probability is presented in Equations (13) and (14).

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}[\mu_{ij}(t)]^{\nu}[\varphi_{ij}(t)]^{\gamma}}{\sum_{j \in allowed_k}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}[\mu_{ij}(t)]^{\nu}[\varphi_{ij}(t)]^{\gamma}}, j \in allowed_k \\ 0, \quad otherwise \end{cases} \tag{13}$$

$$\varphi_{ij}(t) = \begin{cases} \chi, \theta_{si}(t) = \theta_{ij}(t) \\ 1 - \chi, \theta_{si}(t) \neq \theta_{ij}(t) \end{cases} \tag{14}$$

where $\varphi_{ij}(t)$ is the smoothing factor. γ is its corresponding weight coefficient, and χ denotes the turning factor. $\theta_{si}(t)$ is angle between the previous node s to the current node i at time t, while $\theta_{ij}(t)$ is the angle between the current node i to the next node j.

3.5. Global Path Smoothing Strategy

To address the issue where the mobile robot is unable to traverse sharp turns due to excessively small angles at turning points along the optimal path, this paper applies cubic Bézier curves to smooth these turns. This approach ensures that the final path better aligns with the actual motion trajectory of the robot. Specifically, starting from the initial point to the destination, every fourth coordinate along the optimal path is selected. Using these as control points, a third-order Bézier curve is generated between each pair of coordinates, as defined in Equation (15).

$$p(t) = (1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t)p_2 + t^3 p_3 \tag{15}$$

where p_0, p_1, p_2, p_3 are the control points of the curve and $t \in [0, 1]$.

3.6. Implementation of NMS-EACO Algorithm

The NMS-EACO algorithm is composed of five components: an A*-guided heuristic function, an adaptive enhanced pheromone update rule, a state transition probability under nonholonomic constraints, a smoothing factor integrated into the state transition probability, and global path smoothing. The overall workflow of the algorithm is outlined in Algorithm 1.

Algorithm 1 NMS-EACO

Initialize the environment model and the parameters of NMS-EACO according to Table 1
for $i = 1$ to n **do**
    Calculate the A*-based heuristic distance from node i to the destination
    Initialize the direction heuristic matrix according to Equation (6)
end for
for $i = 1$ to n **do**
    for $j = 1$ to map size **do**
        Compute horizontal and vertical coordinate differences for node j
        Calculate the nonholonomic constrained transition probability according to Equation (11)
        Initialize the adaptive pheromone matrix according to Equations (7) and (9) and (10)
    end for
end for
for $k = 1$ to $MaxIterations$ **do**
    for $m = 1$ to $NumberOfAnts$ **do**
        Place ant m at the starting point S
        while ant m has not reached the goal point E **do**
            Select next node based on transition probability integrating:
                – heuristic matrix according to Equation (5)
                – pheromone matrix according to Equation (8)
                – smoothness factor according to Equation (14)
                – nonholonomic constraints according to Equation (12)
        end while
    end for
    for $i = 1$ to $NumberOfPaths$ **do**
        Apply elitist selection and roulette wheel selection to choose paths according to Equation (13)
    end for
    Apply global path smoothing strategy to all selected paths according to Equation (15)
end for
Output the optimal smoothed path

4. Simulation Experiments and Analysis

This section presents simulation experiments of the NMS-EACO algorithm using a software platform. First, the algorithm's parameters are configured. Next, an ablation study is performed to assess the contribution of each component. Finally, NMS-EACO is compared against four other algorithms across three maps of varying sizes.

4.1. Algorithm Parameters Configuration

Parameter settings plays a crucial role in influencing the algorithm's convergence speed, path quality, and overall performance. Drawing on practical engineering experience, this study conducted parameter tuning experiments under various map sizes and obstacle

distributions. Particular attention was given to iterative adjustments of key parameters, such as the pheromone evaporation rate and heuristic function factor. After multiple rounds of evaluation based on metrics like path length and convergence speed, an optimal set of parameters demonstrating stable and high performance was identified, as summarized in Table 1.

Table 1. Parameter settings used in the proposed algorithm.

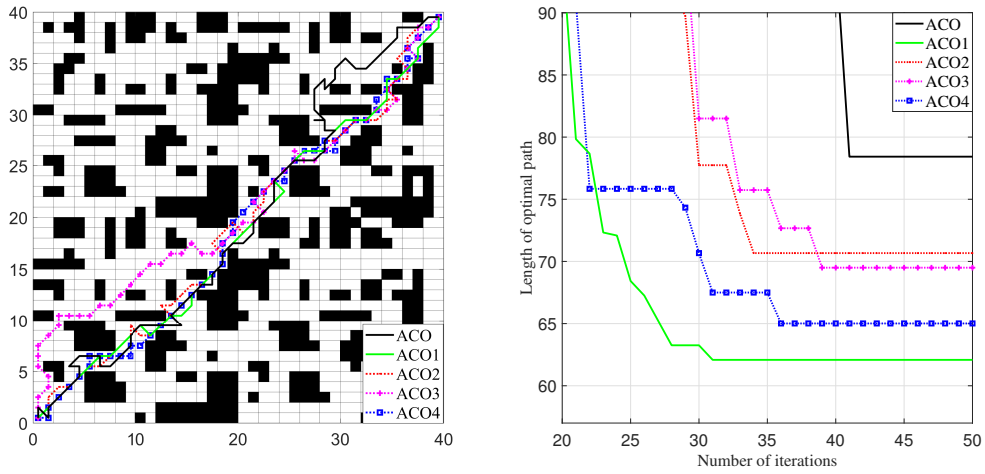
Experimental Parameters	**Parameter Values**
Initial Pheromone Value τ	1
Number of Ants $\boldsymbol{m}$	50
Maximum Number of Iterations $\boldsymbol{N\_max}$	50
Heuristic Function Factor α	1.5
Pheromone Evaporation Factor β	0.5
Pheromone Enhancement Coefficient $\boldsymbol{Q}$	20
Smoothing Factor Weight Coefficient γ	0.6
Turning Factor χ	0.7
Angle Factor ν	0.6

4.2. Ablation Study

To evaluate the effectiveness of each individual mechanism—namely the A*-guided heuristic function, adaptive enhanced pheromone update rule, state transition probability under nonholonomic constraints, and the smoothing factor for optimizing state transitions—this study conducts a series of ablation experiments. Each mechanism is integrated with the basic ACO algorithm to create four extended variants, referred to as ACO1, ACO2, ACO3, and ACO4, as outlined in Table 2. A 40×40 grid map is used as the simulation environment, and each algorithm is executed 10 times. The experimental results are presented in Table 3.

Table 2. Ablation Study Schemes.

Model	**Integrated Mechanism**
ACO1	ACO + A*-guided heuristic function
ACO2	ACO + adaptive enhanced pheromone update rule
ACO3	ACO + state transition probability under nonholonomic constraints
ACO4	ACO + smoothing factor to optimize state transition probability



(**a**) (**b**)

Figure 5. Results of Five ACOs in Ablation Experiments: (**a**) Optimal Paths. (**b**) Convergence Curves of Optimal Paths.

Table 3. Statistical Results of Five ACOs in Ablation Experiment.

Model	Optimal path length	Average path length	Turn times	Mean of convergence generation
ACO	78.43	88.44	51	41
ACO1	62.08	66.04	23	34
ACO2	70.67	79.22	39	33
ACO3	69.5	73.53	37	38
ACO4	65.01	67.59	17	36

As shown in Figure 5 and Table 3, all five ACOs successfully complete the path planning task. The minimum number of iterations for ACO2 through ACO5 to converge to the optimal path is lower than that of the basic ACO. Specially, ACO1 achieves both a shorter optimal path length and the average path length compared to ACO (62.08 < 78.43, 66.04 < 88.44, respectively), with fewer iterations (34 < 41), which proves the effectiveness of the A*-guided heuristic function. ACO2 also shows improvements, with shorter optimal and average path lengths (70.67 < 78.43 and 79.22 < 88.44) and reduced iterations (33 < 41), confirming the benefit of the adaptive enhanced pheromone update rule. Similarly, ACO3 outperforms the basic ACO in optimal and average path lengths (69.5 < 78.43 and 73.53 < 88.44) and requires fewer iterations (38 < 41), validating the use of state transition probability under nonholonomic constraints. In ACO4, the number of turns is significantly reduced (17 < 51), alongside shorter optimal and mean path lengths (65.01 < 78.43 and 67.59 < 88.44), which supports the effectiveness of the smoothing factor for optimizing state transitions. From these results, it is evident that each individual mechanism substantially contributes to enhancing the performance of the ACO algorithm.

4.3. Simulation Experiments Across Three Different Environments

To demonstrate the superiority and applicability of NMS-EACO, simulation experiments are performed on grid maps of three different sizes: 30 × 30, 40 × 40, and 50 × 50. Besides varying map sizes, the obstacle distributions also differ. These three maps are referred to as Map-1, Map-2, and Map-3, respectively, as detailed in Table 4. Additionally,

four ACO variants are selected for experimental comparison, listed in Table 5. To minimize experimental errors and ensure result reliability, each algorithm is executed 30 times.

This study uses optimal path length, average path length, and number of turns to assess path quality. Average convergence generation is to evaluate convergence performance, and standard deviation to measure algorithm stability. Based on practical engineering considerations, path quality is prioritized as the primary evaluation criterion, followed by convergence performance as the secondary criterion, and stability as the tertiary criterion.

Table 4. Map Abbreviation Correspondence

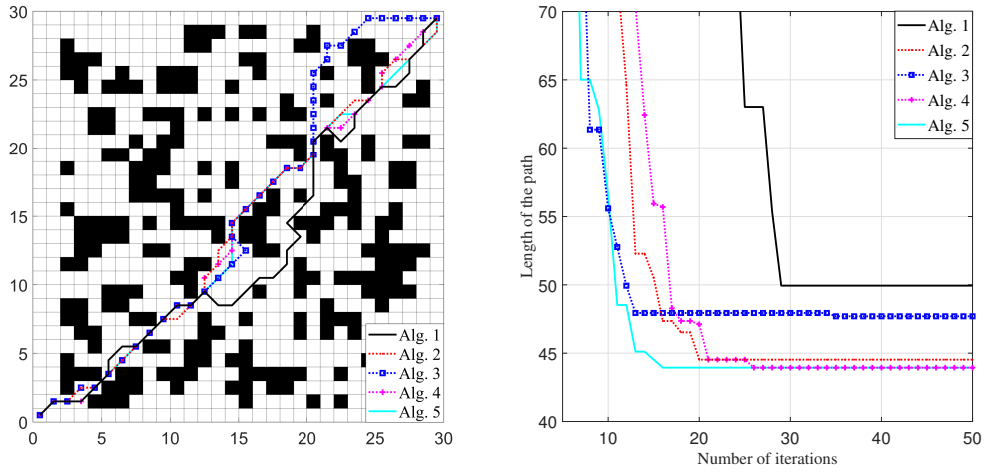
Map size	Map Abbreviation
30×30	Map-1
40×40	Map-2
50×50	Map-3

Table 5. Algorithm Abbreviation Correspondence

Algorithm Name	Algorithm Abbreviation
ACO	Alg.1
The algorithm proposed in [18]	Alg.2
The algorithm proposed in [28]	Alg.3
The algorithm proposed in [29]	Alg.4
NMS-EACO	Alg.5

4.3.1. Experimental Results on Map-1

Figure 6 illustrates the optimal paths and iteration curves of the five algorithms on Map-1. The results from 30 runs of each algorithm were recorded and analyzed, with the statistical summary presented in Table 6.



(**a**) (**b**)

Figure 6. Experimental Results of Five Algorithms on Map-1: (**a**) Optimal Paths. (**b**) Convergence Curves of Optimal Paths.

Table 6. Statistical Results of Five Algorithms on Map-1.

Algorithm	Optimal path length	Mean path length	Mean number of turns	Mean of convergence generation	Path standard deviation
Alg.1	49.94	56.52	32.27	29.03	4.11
Alg.2	44.52	45.02	23.00	21.00	0.74
Alg.3	47.70	48.39	16.53	34.50	0.67
Alg.4	43.94	45.11	18.50	25.30	0.93
Alg.5	43.94	44.46	16.87	17.53	0.63

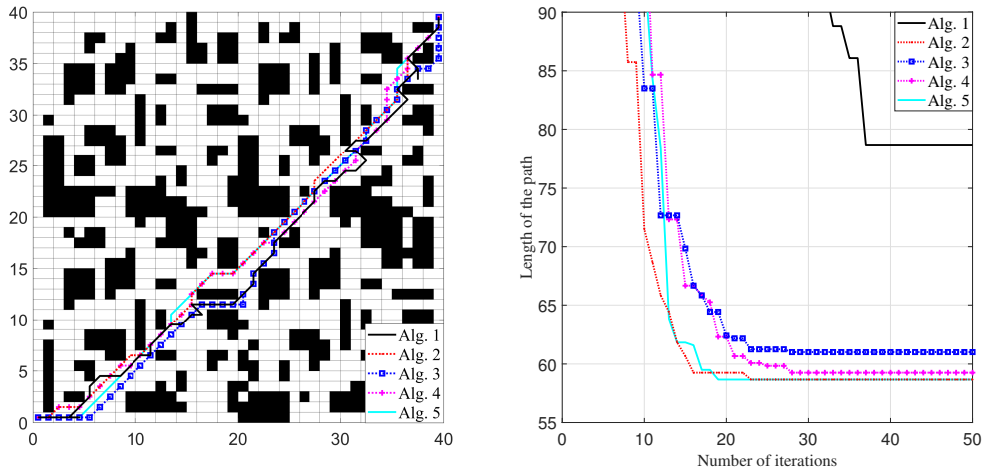
As shown in Figure 6, all five algorithms successfully complete the optimal path planning. Among them, Alg.2, Alg.4, and Alg.5 produce similar path shapes, while the paths generated by Alg.1 and Alg.3 are significantly longer. In the iteration curves, Alg.2 through Alg.5 converge in fewer iterations compared to Alg.1.

According to the data in Table 6, Alg.4 and Alg.5 achieve the shortest optimal path length 43.94, while Alg.2 follows closely at 44.52. The average path lengths for these three algorithms are also comparable. In terms of turn count, Alg.4 is 18.50 and Alg.5 is 16.87, that have fewer turns than Alg.2 as 23, indicating smoother paths. Based on the mean number of convergence iterations, ranked from fewest to most, the order is Alg.5 < Alg.2 < Alg.4, suggesting that Alg.5 has the fastest convergence, followed by Alg.2 and then Alg.4. Additionally, when comparing the standard deviation of path lengths, Alg.5 exhibits the smallest variation at 0.63, followed by Alg.2 at 0.74 and Alg.4 at 0.93, indicating higher stability in path planning.

Therefore, on Map-1, Alg.4 and Alg.5 demonstrate better performance in terms of path quality, convergence speed, and stability, with Alg.2 performing slightly behind them.

4.3.2. Experimental Results on Map-2

Figure 7 displays the optimal paths and iteration curves of the five algorithms on Map-2. The results from 30 runs of each algorithm were collected and analyzed, with the statistical outcomes summarized in Table 7.



(**a**) (**b**)

Figure 7. Experimental Results of Five Algorithms on Map-2: (**a**) Optimal Paths. (**b**) Convergence Curves of Optimal Paths

Table 7. Statistical Results of Five Algorithms on Map-2

Algorithm	Optimal path length	Mean path length	Mean number of turns	Mean of convergence generation	Path standard deviation
Alg.1	78.67	86.81	50.93	35.36	11.16
Alg.2	58.66	60.85	20.40	23.63	1.92
Alg.3	61.01	64.77	17.33	27.36	2.24
Alg.4	58.66	61.10	21.53	21.53	1.54
Alg.5	58.66	59.98	12.50	19.40	1.00

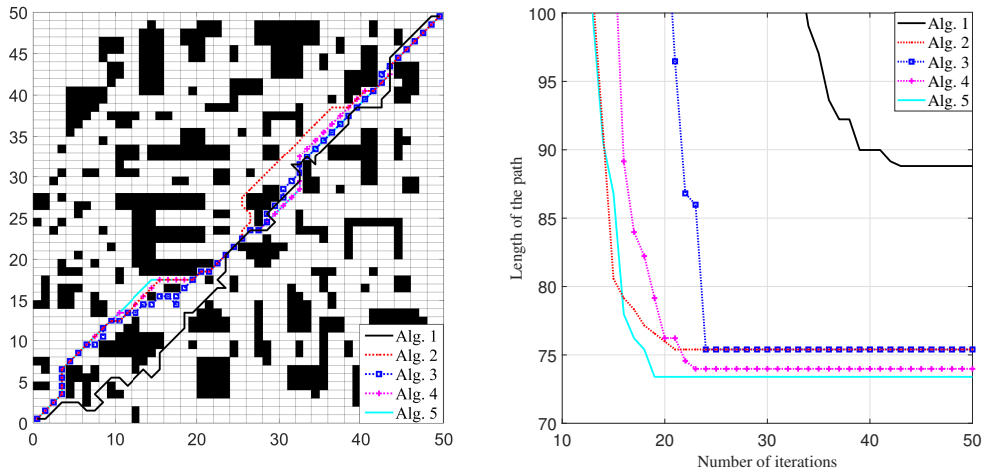
As shown in Figure 7, all five algorithms have completed the optimal path planning task, with the resulting paths displaying similar shapes. However, the path generated by Alg.1 contains more turning points and appears less smooth. The iteration curves for the optimal paths of Alg.2 through Alg.5 follow similar trends and converge with noticeably fewer iterations compared to Alg.1.

According to Table 7, Alg.2, Alg.4, and Alg.5 achieve the shortest optimal path length at 58.66, and their average path lengths are also closely aligned. Among them, Alg.5 has the fewest turns (12.50), compared to Alg.2 (20.04) and Alg.4 (21.53), indicating a smoother path. In terms of average convergence iterations, Alg.5 (19.40) performs the best, followed closely by Alg.4 (21.53), both of which outperform Alg.2 (23.63). Regarding path stability, Alg.5 again shows the smallest standard deviation (1.00), followed by Alg.4 (1.54), indicating more consistent performance.

Overall, in Map-2, Alg.5 and Alg.4 deliver the best results across all three evaluation metrics: path quality, convergence speed, and stability, which are followed by Alg.3.

4.3.3. Experimental Results on Map-3

Figure 8 presents the optimal paths and iteration curves of the five algorithms on Map-3. Each algorithm was executed 30 times, and the resulting data were recorded and analyzed. The statistical results are summarized in Table 8.



(**a**) (**b**)

Figure 8. Experimental Results of Five Algorithms on Map-2: (**a**) Optimal Paths. (**b**) Convergence Curves of Optimal Paths.

Table 8. Statistical Results of Five Algorithms on Map-3.

Algorithm	Optimal path length	Mean path length	Mean number of turns	Mean of convergence generation	Path standard deviation
Alg.1	88.81	106.85	64.50	38.83	11.26
Alg.2	75.39	77.58	30.36	29.20	1.29
Alg.3	75.44	77.70	22.43	26.30	1.74
Alg.4	73.98	76.81	19.50	26.33	1.88
Alg.5	73.39	74.01	13.33	20.46	0.61

As illustrated in Figure 8, the optimal paths generated by Alg.2 through Alg.5 are quite similar, while the path produced by Alg.1 appears longer and less smooth. In the iteration curves, Alg.2, Alg.4, and Alg.5 converge more quickly than Alg.1 and Alg.3, with Alg.3 requiring fewer iterations than Alg.1.

According to Table 8, Alg.5 achieves both the shortest optimal path length (73.89) and mean path length (74.01), followed by Alg.4 (73.98, 76.81), Alg.2 (75.39, 77.58), and Alg.3 (75.44, 77.70). Additionally, Alg.5 records the lowest average number of turns (13.3), compared to Alg.4 (19.50), Alg.3 (22.43), and Alg.2 (30.36), indicating a smoother path. When comparing average convergence iterations, Alg.5 (20.46) again performs best, followed by Alg.3 (26.30) and Alg.4 (26.33). In terms of stability, measured by the standard deviation of path lengths, Alg.5 shows the smallest variation (0.61), with Alg.3 (1.74) and Alg.4 (1.88) following.

In summary, for Map-3, Alg.5 delivers the strongest overall performance across all evaluation metrics, followed by Alg.4 and then Alg.3.

Table 9. The Mean of Differences in Five Metrics between Alg.4 and Alg.5 across the Three Maps.

	Optimal path length	Mean path length	Mean number of turns	Mean of convergence generation	Path standard deviation
Differences	0.02	1.53	5.61	5.25	0.70

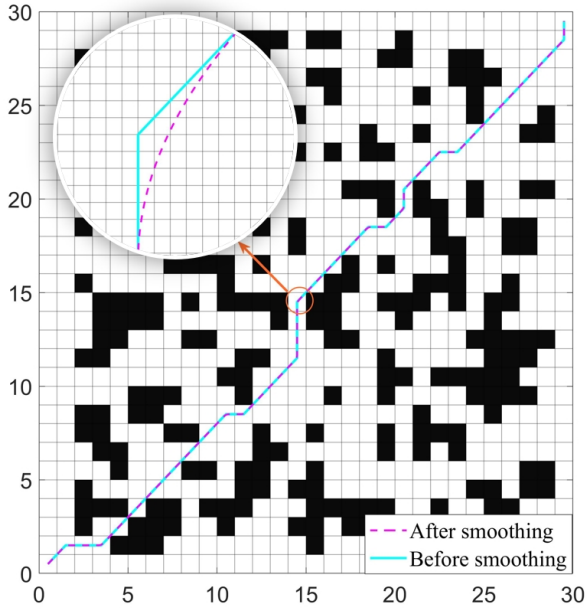


Figure 9. Path Smoothed Using a Cubic Bézier Curve

In summary, on Map-1, Alg.2, Alg.4, and Alg.5 demonstrate better performance, while on Maps 2 and 3, Alg.3 through Alg.5 perform more effectively. Although the performance of each algorithm is influenced by the specific map, both Alg.4 and Alg.5 show strong adaptability across all three environments. As detailed in Table 9, Alg.4 and Alg.5 have comparable optimal path lengths. However, Alg.5 achieves a mean path length that is 1.53 units shorter than Alg.4, converges 5.25 iterations faster on average, and has a 0.7 lower standard deviation in path length, highlighting the advantages of the A*-guided heuristic function and the adaptive enhanced pheromone update rule. Moreover, the average number of turns for Alg.5 is 5.61 fewer than that of Alg.4, further demonstrating the effectiveness of the state transition probability under nonholonomic constraints and the smoothing factor. Additionally, as illustrated in Figure 9, the path optimized using a cubic Bézier curve is smoother and better aligned with the mobile robot's turning characteristics.

5. Conclusion

This paper addresses the limitations of traditional ACO in mobile robot path planning and introduces the NMS-EACO algorithm as an improved solution. To tackle issues such as slow convergence and the tendency to fall into local optima, the A*-guided heuristic function and adaptive enhanced pheromone update rule are proposed. To further address problems related to excessive turning points and poor path smoothness, a state transition probability model incorporating nonholonomic constraints is developed, along with a smoothing factor to optimize state transitions. Additionally, a cubic Bézier curve is applied to refine the global path. Through a series of ablation studies and simulation experiments across various map scenarios, the results demonstrate that NMS-EACO achieves faster convergence, stronger global search capability, and significantly improved path smoothness.

Author Contributions: C.Z.: conceptualization, methodology, data curation, validation; J.Xu.: writing—original draft preparation, software; C.G.: data curation, software; X.W. methodology, validation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Shandong Provincial Natural Science Foundation of China (ZR2024QF303), the major innovation projects of science, education, industry integration of Qilu University of Technology (Shandong Academy of Science) (2024ZDZX09), and the National Key Research and Development Program of China (2023YFB3711400).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors have no external support to acknowledge in this section, including administrative, technical assistance, or donations in kind. Also, no generative artificial intelligence tools were utilized during the research and manuscript preparation process.

Conflicts of Interest: The authors declare no potential conflicts of interest with respect to the research,authorship, and/or publication of this article.

References

1. Du, Y.; Xiong, Y.; Chen, M. Physics-Embedded Motion Planning With Contact Handling for Continuum Surgical Robots. *IEEE Robotics and Automation Letters* **2025**, *10(7)*, 7031–7038.
2. Sari, D. W.; Dwijayanti, S.; Suprapto, B. Yudho. Path Planning for an Autonomous Vehicle based on the Ant Colony Algorithm: A Review. *2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP)* **2023**, 57–62.
3. Tao, B.; Kim, J. H.; Shen, Y. Mobile robot path planning based on bi-population particle swarm optimization with random perturbation strategy. *Journal of King Saud University-Computer and Information Sciences* **2024**, *36(2)*, 101974.
4. Sun, W.; Yuan, Y.; Gao, H. Hierarchical Control for Partially Feasible Tasks With Arbitrary Dimensions: Stability Analysis for the Tracking Case. *IEEE Transactions on Automatic Control* **2024**, *69(9)*, 5883–5898.
5. Sun, W.; Yuan, Y. Passivity based hierarchical multi-task tracking control for redundant manipulators with uncertainties. *Automatica* **2023**, *155*, 111159.
6. Katona, K.; Neamah, H. A.; Korondi, P. Obstacle avoidance and path planning methods for autonomous navigation of mobile robot. *Sensors* **2024**, *24(11)*, 3573.
7. Huang, T.; Pan, H.; Sun, W. Sine Resistance Network-Based Motion Planning Approach for Autonomous Electric Vehicles in Dynamic Environments. *IEEE Robotics and Automation Letters* **2022**, *8(2)*, 2862–2873.
8. Gherold, V.; Mandralis, I.; Sihite, E. Self-Supervised Cost of Transport Estimation for Multimodal Path Planning. *IEEE Robotics and Automation Letters* **2025**, *10(7)*, 6872–6879.
9. Wang, H.; Wang, W.; Shen, Y. A Kinematic Constrained Batch Informed Trees Algorithm With Varied Density Sampling for Mobile Robot Path Planning. *IEEE Robotics and Automation Letters* **2025**, *10*, 6912–6919.
10. Wang, t.; Ji, C.; Wang, X. A vehicle path planning method for time-delay optimization and energy saving in edge computing. *Engineering Journal of Wuhan University* **2024**, *57(11)*, 1619–1626.
11. Eido, W. M.; Ibrahim, I. M. Ant colony optimization (ACO) for traveling salesman problem: a review. *Asian Journal of Research in Computer Science* **2025**, *18(2)*, 20-45.
12. Li, W.; Wang, G. Research on Multi-Agent Path Planning Based on lmproved Ant Colony Algorithm. *Journal of Jilin University(Information Science Edition)* **2024**, *42(04)*, 654–661.
13. Yang, G.; Fu, S.; Liu, Z. Path planning of indoor mobile robot based on improved ant colony algorithm. *Science Technology and Engineering* **2019**, *19(19)*, 175–179.
14. Yang, H. UAV3D flight path planning method based on improved antcolony andartifcial potential field method. *Hebei University* **2023**.
15. Shao, Q.; Shi, W. Research on robot path planning based on improved ant colony algorithm. *Moderm Manufacture Engineering* **2023**, *06*, 46–51.
16. Yuan, W.; You, X.; Liu, S. Two-population ant colony algorihm based on dynamic learning mechanism. *Journal of Computer Science and Exploration*. **2019**, *13(07)*, 1239–1250.
17. Chen, Y.; Gao, M. Ant colony algorithm based on initial pheromone allocation and dynamic updating. *Computer Engineering and Applications*. **2022**, *58(02)*, 95–101.
18. Sun, R.; Zhang, W. Smooth path planning of mobile robot based on improved ant colony algorithm. *Journal of Graphics*. **2019**, *40(2)*, 6912–6919.
19. Wu, S.; Dong, A.; Li, Q. Application of ant colony optimization algorithm based on farthest point optimization and multi-objective strategy in robot path planning. *Applied Soft Computing* **2024**, *167*, 112433.
20. Ye, K.; Zhang, C.; Ning, J. Ant colony algorithm with a strengthened negative feedback mechanism for constraint-satisfaction problems. *Information Sciences*. **2017**, *406–407*, 29–41.
21. Wang, D.; Liu, Q.; Yang, J.; Zhang, H. Research on path planning for intelligent mobile robots based on improved A* algorithm. *Symmetry*. **2024**, *16(10)*, 1311.

22. Liu, S.; Huang, Y. Application of multi-strategy ant colony algorithm in robot path planning. *Computer Engineering and Applications* **2022**, *58(06)*, 278–286.
23. Zuo, S.; Mao, Z.; Fan, C. Dynamic planning of crowd evacuation path for metro station based on Dynamic Avoid Smoke A-Star algorithm. *Tunnelling and Underground Space Technology*. **2024**, *154*, 106145.
24. Abdullah, A.; Mustafa, U. Angle-based multi-goal ordering and path-planning using an improved A-star algorithm. *Robotics and Autonomous Systems*. **2025**, *190*, 105001.
25. Yuan, Y.; Sun, W. Continuously Shaping Prioritized Jacobian Approach for Hierarchical Optimal Control With Task Priority Transition. *IEEE Transactions on Robotics* **2025**, *41*, 1639–1656.
26. Huang, T.; Wang, J.; Pan H. Adaptive bioinspired preview suspension control with constrained velocity planning for autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*. **2023**, *8(7)*, 3925-3935.
27. Huang, T.; Wang, J.; Pan H. Finite-time fault-tolerant integrated motion control for autonomous vehicles with prescribed performance. *IEEE Transactions on Transportation Electrification*. **2022**, *9(3)*, 4255-4265.
28. Fang, S.; Deng, Z.; Li, p. Improved strategy of ant colony optimization for path planning via stochastic pheromone updating and cyclic initialization.*Journal of Mechanical Science and Technology*. **2025**, *39(4)*, 2051-2062.
29. Fu, B.; Chen, Yuming.; Quan, Yi. Bidirectional artificial potential field-based ant colony optimization for robot path planning. *Robotics and Autonomous Systems*. **2025**, *183*, 104834.
30. Ozdemir, K.; Tuncer, A. Navigation of Autonomous Mobile Robots in Dynamic Unknown Environments Based on Dueling Double Deep Q Networks. *Engineering Applications of Artificial Intelligence* **2025**, *139*, 109498.
31. Tang, Y.; Zakaria, M. A.; Younas, M. Path planning trends for autonomous mobile robot navigation: A review. *Sensors* **2025**, *25(4)*, 1206.
32. Huang, J.; Chen, C.; Shen, J.; Liu, G.; Xu, F. A Self-Adaptive Neighborhood Search A-Star Algorithm for Mobile Robots Global Path Planning. *Computers and Electrical Engineering* **2025**, *123*, 110018.
33. Sun, Z.; Xia, B.; Xie, P.; Li, X.; Wang, J. NAMR-RRT: Neural Adaptive Motion Planning for Mobile Robots in Dynamic Environments. *IEEE Transactions on Automation Science and Engineering* **2025**, *22*, 13087-13100.