A Variable Size Grid Based Ant Colony Optimization Algorithm for Robot Path Planning

Xu Guo, Wenlong Ji, Siquan Li, Anqi Xu, Yanling Shang, and Fangzheng Gao

***Abstract*—In this study, we propose an enhanced Ant Colony Optimization algorithm embedded within a variable-resolution grid framework to improve the performance of mobile robot path planning. To address the typical limitations of conventional ACO methods—namely, their slow convergence, tendency to get trapped in local optima, and inability to produce smooth trajectories—the algorithm incorporates a series of refined strategies grounded in grid-based representation.Firstly, an adaptive grid partitioning strategy with variable resolution is developed: it enlarges the search neighborhood in complex, obstacle-dense regions and contracts it in open areas, thereby achieving a balance between search efficiency and path quality. Secondly, a refined heuristic function, drawing inspiration from the A* algorithm, is employed to guide ants more effectively toward the goal. The A* algorithm is also leveraged to initialize the pheromone distribution, which helps accelerate convergence during the early iterations. Moreover, a directional bias mechanism is integrated to suppress redundant node exploration and improve search efficiency. To further enhance the algorithm's global exploration capability and avoid premature convergence, a dynamic pheromone update scheme is proposed—featuring both a reward-penalty model and an adaptive evaporation rate.Comprehensive simulations carried out in MATLAB evaluate the performance of the proposed approach in comparison with the standard ACO, various enhanced ACO versions, and other intelligent optimization techniques, including Genetic Algorithm , Particle Swarm Optimization, and Differential Evolution. The results consistently demonstrate that the modified algorithm achieves more efficient convergence, produces shorter and smoother paths, and exhibits greater robustness—highlighting its strong potential for practical applications and future hardware integration.**

***Index Terms*—ACO algorithm, path planning, variable-size grid, A* algorithm, pheromone update.**

I. Introduction

With the rapid advancements in artificial intelligence technology, mobile robots have achieved widespread deployment and application in various fields such as logistics and warehousing, autonomous driving, medical services, and agricultural production. These diverse application scenarios have imposed stricter requirements on the autonomous navigation technology of robots. Among them, path planning technology, as one of the core elements, directly affects the robots' motion efficiency, the smoothness of their paths, and their adaptability in complex and ever-changing environments. However, when confronted with high-dimensional, dynamic, and intricate real-world environments, traditional path planning algorithms still face numerous challenges that urgently need to be addressed.

Path planning serves as a fundamental research focus in the field of robotics, particularly playing a pivotal role in warehouse automation and intelligent logistics systems. The primary objective of path planning is to determine an optimal trajectory on a given environmental map—one that ensures safety, avoids collisions, and minimizes the overall travel distance between a specified start and target location, all while adhering to predefined optimization criteria. In recent years, researchers both at home and abroad have devoted considerable efforts to the study of global path planning methods. These approaches range from classical deterministic algorithms, including Dijkstra's algorithm, the A* search strategy, and the artificial potential field method, to a variety of intelligent optimization techniques such as particle swarm optimization (PSO), artificial neural networks (ANN), genetic algorithms (GA), and ant colony optimization (ACO). The integration of these traditional and heuristic algorithms has significantly advanced the performance and adaptability of robotic path planning in dynamic and complex environments.

To address the shortcomings of current algorithms, scholars have been actively investigating innovative path planning strategies that not only boost computational efficiency and path accuracy but also strengthen the autonomous navigation capabilities of mobile robots operating in intricate and dynamic environments.

Anil Kumar et al. [1] introduced the ACO-UCR algorithm to improve energy efficiency in clustered wireless sensor networks, demonstrating its advantages in wireless communication scenarios. Cai et al. proposed a two-dimensional ant colony optimization method that integrates heuristic strategies with the firefly algorithm to enhance adaptability across different application environments [2]. Kale Ishaan et al. developed the ACO-CI algorithm by combining ant colony optimization with intelligent queue mechanisms to address constraint-related engineering challenges, thereby lowering computational overhead [3]. Hui et al. designed an enhanced ACO model incorporating wolf pack distribution techniques to tackle route planning issues more effectively [4]. For multi-UAV systems, Athira et al. proposed the ACO-DTSP algorithm, which refines pheromone update rules to improve swarm path planning performance [5]. Liu et al. [6] intro-

Manuscript received April 28, 2025. revised July 15, 2025. This work was partially supported by the National Natural Science Foundation of Jiangsu Province under grant BK20201469.

Xu Guo is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (e-mail:y00450240314@njit.edu.cn).

Wenlong Ji is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (e-mail:y00450240413@njit.edu.cn).

Siquan Li is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (email: y00450230122@njit.edu.cn).

Anqi Xu is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (e-mail:y00450240304@njit.edu.cn).

Yanling Shang is an associate professor in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (corresponding author, e-mail: hnnhsyl@126.com.com).

Fangzheng Gao is a professor in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (e-mail:gaofz@126.com).

duced an improved MPGA-ACO-BP hybrid approach, which strengthens evaluation accuracy in data fusion tasks and demonstrates practical applicability. Zhu et al. [7] presented a variable-resolution grid-based ACO method that accelerates early-stage pheromone deposition, helping to mitigate the risk of premature convergence. Jai et al. put forward an intelligent dynamic routing optimization framework that synergizes ACO with genetic algorithms to enhance route precision [8]. To resolve issues such as deadlock, limited search capability, and susceptibility to local optima in traditional ACO algorithms, Si et al. proposed a novel parallel ACO variant [9]. Lastly, Shi et al. [10] introduced the self-adaptive LIEAD algorithm, which utilizes local information entropy to address distributed constraint optimization problems with improved adaptability and performance.

Ant Colony Optimization , inspired by the collective foraging behavior observed in real ant colonies, is known for its robust global search capability and has been extensively utilized in solving path planning problems. Despite its widespread application, conventional ACO algorithms still suffer from several limitations, including low computational efficiency, a tendency to fall into local optima, and the tendency to produce paths that lack smoothness. To overcome these challenges, numerous enhancement strategies have been introduced by researchers—ranging from refining the state transition rules and improving pheromone update strategies to integrating ACO with other metaheuristic algorithms. Nevertheless, challenges remain, particularly in accelerating the search process during the early phases and enhancing the overall effectiveness of the generated paths [11]. In response, this study puts forward an improved ACO framework that incorporates variable grid modeling, refined pheromone initialization, and adaptive pheromone update mechanisms, with the aim of boosting performance in both convergence speed and path quality.

II. Fundamentals of Ant Colony Optimization Algorithm

A. The Current Situation of Models and Algorithms

Global path planning relies on environmental modeling methods such as visibility graphs, free-space representations, and grid-based approaches. In this study, the grid-based method is adopted as a representative example, with its conceptual model illustrated in Figure 1.

The grid approach discretizes the robot's workspace into uniform grid cells, each encoded with binary information to indicate either free space or obstacle presence. Common representations include quadtree (2D) and octree (3D) structures, which provide flexible hierarchical decomposition of the environment. Once the environment is encoded into grids, various optimization algorithms can be employed to search for feasible paths.

The resolution of the grid model—defined by the size of each grid cell—has a direct impact on both the richness of environmental information captured and the computational efficiency of the planning process. A coarser grid (i.e., larger cells) reduces memory usage and shortens planning time but sacrifices spatial accuracy and may fail to identify viable paths in cluttered or narrow spaces. Conversely, a finer grid increases resolution and improves pathfinding capability in dense or obstacle-rich environments, but at the cost of greater computational burden and memory demand.

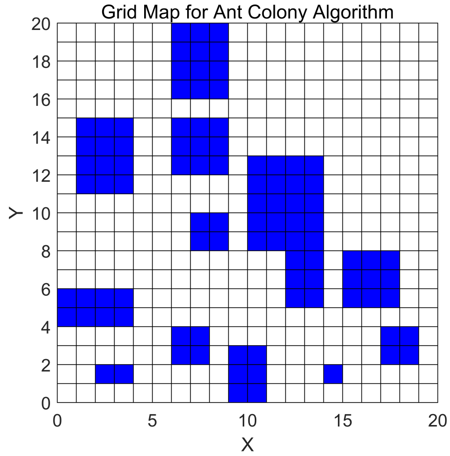


Fig. 1. Grid Method Model.

Fig. 1 shows a grid-based model of a 2D space for path - planning environment representation. The axes range from 0 to 20, and the area is divided into equal - sized square grid cells. Blue squares indicate obstacles or non - traversable areas, while white squares represent free or traversable space. This model discretizes the environment, making it easier to apply path - planning algorithms. Note that grid size affects planning accuracy and efficiency: larger cells reduce accuracy, while smaller cells provide higher accuracy but boost computation and storage demands.

B. State Transition Dynamics

Within the fundamental framework of the Ant Colony Optimization algorithm, the movement probability of an ant from node i to a subsequent node j is governed by the state transition rule defined as follows:

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}\,[\eta_{ij}(t)]^{\beta}}{\sum_{s\in\text{ allow}}[\tau_{ij}(t)]^{\alpha}\,[\eta_{ij}(t)]^{\beta}}, & s\in\text{allow}_k\\ 0, & s\notin\text{allow}_k\end{cases}$$
$$n_{ij}=\frac{1}{d_{ij}}$$
$$d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \tag{1}$$

Here, τ_{ij} denotes the pheromone level associated with the edge connecting node i to node j, while η_{ij} represents the heuristic desirability of that path. The parameters α and β control the relative influence of pheromone intensity and heuristic information on the transition probability, respectively. The term d_{ij} indicates the Euclidean distance between nodes i and j, calculated using their coordinates (x_i, y_i) and (x_j, y_j). Additionally, allow$_k$ refers to the set of feasible paths available to ant k at the upcoming decision stage.

Ant Colony Optimization is an evolutionary algorithm inspired by the natural foraging behavior of real ant colonies. It employs a probabilistic search mechanism to identify optimal or near-optimal paths within a graph structure.The core principle of ACO is grounded in the indirect communication among ants through pheromone deposition. In natural settings, ants collectively explore their environment for food sources. During this exploration, they deposit a chemical substance known as pheromone along their paths. As time progresses, the pheromone trails begin to dissipate due to natural evaporation. However, paths with higher pheromone concentrations become more attractive to other ants, reinforcing those routes through a positive feedback mechanism. This iterative process leads to a higher likelihood that ants will converge on the shortest route, as shorter paths accumulate pheromone more quickly and retain it more effectively, thus guiding subsequent ants toward optimal solutions.

The ACO algorithm primarily operates through two fundamental mechanisms: state transition and pheromone updating.

$$\begin{gathered}\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \\ \Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)\end{gathered} \tag{2}$$

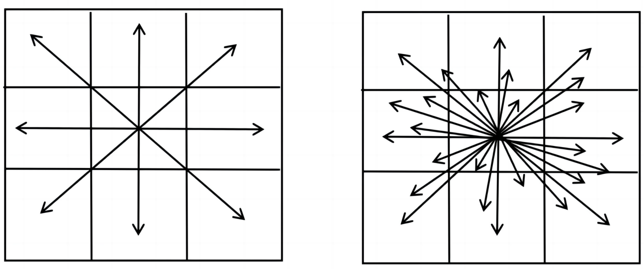
where m represents the total number of ants in each iteration; ρ denotes the pheromone evaporation coefficient; and $\Delta\tau_{ij}(t)$ indicates the amount of pheromone deposited on the path from node i to node j at time t.

Obtaining the optimal solution: Each ant selects the next node j from node i according to the transition probability P_{ij}. After all ants complete one iteration , the pheromone levels on all paths are updated accordingly. This iterative process repeats until a predefined number of cycles is reached. Ultimately, the majority of ants converge on the same path, which is regarded as the shortest path.

III. Advanced Ant Colony Optimization Method

A. Variable Grid

In path planning, the traditional ant colony algorithm has 8 search directions, represented by 8 arrows, with the robot's minimum turning angle being 90°. Wu et al. improved the algorithm [12], changing the 8 domain search to a 48 domain search. Now, the robot has 32 search directions and a minimum turning angle of 11.25°, enhancing path smoothness. But the 7×7 neighborhood search increases nodes, thus increasing search time.



(a) 3×3Domain Search (b) 7×7Domain Search

Fig. 2. Neighborhood Structure of Variable-size Grid.

This chapter introduces a variable grid ant colony algorithm. When there are no obstacles around, Fig. 2 employs a 3×3 search neighborhood for rapid searching. When encountering obstacles, it adaptively switches to a 7×7 search neighborhood to reduce waypoints and achieve a smoother path. After passing through obstacles, it reverts to a 3×3 search neighborhood to accelerate searching. This method saves search time and yields better search results [7].

B. A* Algorithm

The A* algorithm is a well-established and effective graph search method extensively applied in areas such as path planning, robot navigation, and game development. It integrates the shortest-path strategy from Dijkstra's algorithm with the speed advantages provided by heuristic search. Through the incorporation of a heuristic function, the algorithm enhances search efficiency without compromising the optimality of the solution.

The core concept of the A* algorithm is to start from the initial node and gradually expand the path nodes that are most likely to reach the goal until an optimal path from the start to the end is found. At every iteration, the algorithm expands the node that minimizes the cost equation $f(n)$. This equation is composed of two components:

$$f(n) = g(n) + h(n) \tag{3}$$

where $g(n)$ represents the cost incurred from the start node to the current node n, and $h(n)$ denotes the heuristic estimate of the cost from node n to the goal node. Consequently, the cost function is defined as

$$f(n) = g(n) + h(n),$$

which estimates the total cost of reaching the goal through node n.

Frequently used heuristic functions include the Euclidean distance, Manhattan distance, and Chebyshev distance. Provided that the heuristic function h(n) is either consistent or admissible, the A algorithm ensures the discovery of an optimal path between the start and goal nodes. Thanks to its strong performance and modular, extensible design, A has become a widely accepted baseline approach in path planning tasks. While it can efficiently compute optimal paths in static scenarios, its significant computational demands hinder real-time application in dynamic or complex environments, often necessitating integration with other techniques or algorithmic enhancements.

C. Improved Heuristic Function

This study takes advantage of the superior optimization capabilities of the A* algorithm and the efficient search process. We incorporated its heuristic-guided search principle into the heuristic function of the ant colony optimization (ACO) algorithm to enhance the discovery of the optimal path. By integrating an estimated movement cost, the ants are directed more rapidly toward the goal. Additionally, our improved heuristic function introduces a curvature suppression operator designed to reduce both the quantity of turns and the total accumulated turning angle. The formulation of the enhanced heuristic function is detailed below:

$$\begin{aligned}
&n_{ij}(t) = \frac{Q}{g(n)+h(n)+C(n)} \\
&C(n) = \phi_1 \times \text{bend} + \phi_2 \times \text{ angle} \\
&g(n) = \sum_{i=1}^{n} S_{(i)} \\
&h(n) = \sqrt{\left(f_x - n_x\right)^2 + \left(f_y - n_y\right)^2}
\end{aligned} \tag{4}$$

Here, Q denotes a constant value greater than 1. The bending inhibition factor is represented by $C(n)$, where bend indicates the number of turns between the previous and the next node. The turning angle is given by angle. Conversion coefficients for the quantity and magnitude of these turns are represented by ϕ_1 and ϕ_2, respectively. The cost from the start node to node n is expressed as $g(n)$. The distance moved from the parent of node i to node $i-1$ is denoted by $S(i)$. Meanwhile, $h(n)$ estimates the cost from node n to the destination. The current node's coordinates are (n_x, n_y), and the target node's position is given by (f_x, f_y).

D. Improvements in Pheromone Update Mechanism

The pheromone concentration in the ant colony algorithm is updated based on equation (1), where ants that traverse longer paths deposit less pheromone along their routes. As pheromones evaporate over time, iterative updates guide the ant colony toward convergence on an optimal path, ultimately yielding a globally optimal solution. To enhance convergence speed and prevent entrapment in local optima, this paper proposes an improved pheromone update strategy. Specifically, after each iteration, pheromone updates are applied only to the paths of ants that successfully reach the destination, while those stuck or deadlocked in the map are excluded. Moreover, a reward–penalty mechanism is introduced for the best-performing path in each iteration: ants whose paths outperform the current global optimum are rewarded with increased pheromone deposition, whereas those that fall short experience a reduction in pheromone levels along their paths. Additionally, the worst-performing paths in each iteration are subjected to penalties, further decreasing pheromone concentration on those routes [13].

$$\begin{aligned} \tau_{ij}(t+1) &= (1-\rho)\tau_{ij}(t) + \frac{1}{\text{now\_best}} - \frac{\text{now\_best}}{\text{best}} \\ \tau_{ij}(t+1) &= (1-\rho)\tau_{ij}(t) + \frac{1}{\text{no worst}} \end{aligned} \tag{5}$$

where now best denotes the optimal solution identified in the current iteration,now worst corresponds to the poorest solution in the same iteration, and best refers to the globally optimal solution found so far.

The pheromone evaporation coefficient, denoted by ρ, causes the pheromone levels on paths to gradually diminish, thereby steering the algorithm toward convergence on the optimal route. This paper introduces an adaptive approach to adjust ρ dynamically in response to different phases of the algorithm's iterations. During the initial stages, ρ is increased to accelerate convergence speed. Conversely, in the middle and later phases, ρ is reduced to strengthen the algorithm's ability to perform global optimization.

The pheromone evaporation coefficient ρ controls the gradual decay of pheromone on the paths, playing a crucial role in guiding the algorithm toward convergence on the optimal solution. In this paper, an adaptive adjustment strategy for ρ is proposed, wherein the coefficient is dynamically modified based on the current stage of algorithm iteration. During the initial phase, a higher value of ρ is adopted to accelerate the convergence process. As the algorithm progresses into the middle and later stages, ρ is gradually reduced to strengthen the algorithm's ability to perform global optimization and avoid premature convergence.

$$\rho(t+1) = \begin{cases} \rho_{\max}, & \rho \geqslant \rho_{\max} \\ \dfrac{2}{1+e^{t-\mathrm{b}}}\rho(t), & \rho_{\min} \leqslant \rho \leqslant \rho_{\max} \\ \rho_{\min}, & \rho \geqslant \rho_{\min} \end{cases} \tag{6}$$

where t represents the current iteration number, $\rho_{\min}$ is the minimum value of the pheromone evaporation coefficient ρ, $\rho_{\max}$ is the maximum value of the pheromone evaporation coefficient ρ, and b is a constant.

E. Dynamic Pheromone Update

The conventional Ant Colony Optimization algorithm updates pheromones based on heuristic rules, which heavily depend on prior knowledge. This reliance can lead to the omission of potentially better solutions, resulting in premature convergence and entrapment in local optima. To address these limitations, this paper introduces a reward-and-penalty-based pheromone update mechanism. This strategy enables more flexible and adaptive pheromone regulation, dynamically reinforcing or suppressing certain paths. As a result, the algorithm's adaptability is enhanced, convergence speed is improved, and the risk of poor solutions is reduced, thereby increasing the likelihood of identifying the global optimum.

Specifically, as expressed in equation (7), the mechanism increases the pheromone concentration on the best-performing path in each iteration, encouraging ants to follow this route in future searches. Conversely, the worst-performing paths are penalized by reducing their pheromone levels, lowering their probability of being chosen. This dynamic and targeted adjustment approach strengthens the algorithm's global search ability while maintaining efficient convergence.

$$\begin{aligned} &\tau_{ij}(k+1) = (1-\rho)\tau_{ij}(k) + \textstyle\sum_{t=1}^{T} \Delta\tau_{ij}^{t}(k) \\ &\Delta\tau_{ij}^{t}(k) = \begin{cases} \dfrac{Q}{L_m} + \dfrac{L_y - L_m}{L_{mean} - L_y}, & L_m \leqslant L_y \\ \dfrac{Q}{L_m} - \dfrac{L_m - L_c}{L_c - L_{mean}}, & L_m \geqslant L_c \\ \dfrac{Q}{L_m}, & \text{Otherwise} \end{cases} \end{aligned} \tag{7}$$

In this context, ρ denotes the pheromone evaporation coefficient, Q represents the pheromone intensity constant, Lm is the total length of the path traversed by ant m in the current iteration, Ly refers to the shortest path identified in the current iteration, Lc corresponds to the longest (worst) path, and Lmean is the average length of all paths during that iteration.

Traditionally, the ant colony optimization algorithm employs a fixed evaporation coefficient ρ to update pheromone levels after each iteration. However, a small ρ value may cause pheromones to accumulate excessively on certain paths, increasing the likelihood of premature convergence to local optima. Conversely, an overly large ρ can result in rapid pheromone decay, thereby weakening the algorithm's guidance and stability.

To overcome this limitation, this paper proposes an adaptive strategy for adjusting the evaporation coefficient. By decreasing ρ progressively as the number of iterations increases, the algorithm reduces the ants' over-reliance on pheromone trails, thereby enhancing the diversity and randomness of path exploration. This dynamic adjustment mechanism is mathematically described by equation (8), and the changing trend of ρ over iterations is illustrated in Fig. 3.

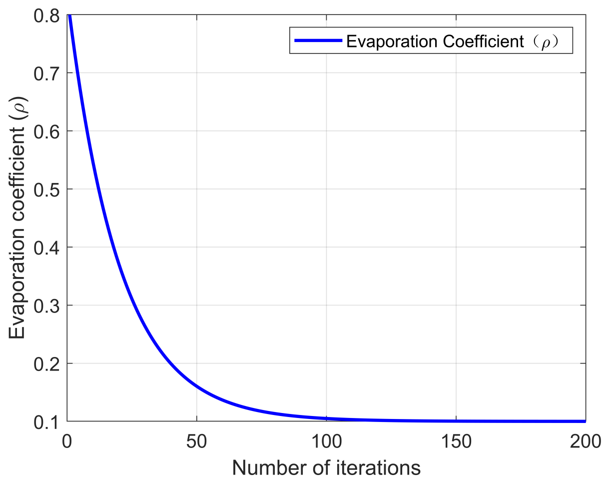


Fig. 3. The Relationship Between the Evaporation Coefficient and the Number of Iterations.

$$\rho(T+1) = \begin{cases} \rho_{\min} & \rho \leqslant \rho_{\min} \\ \dfrac{T_{\max}}{T_{\max}+3T} * \dfrac{1}{e^{1-\rho(T)}} & \text{else} \end{cases} \tag{8}$$

Here, r denotes the coefficient of evaporation, T refers to the current iteration index, $T_{\max}$ specifies the maximum allowed number of iterations, and $\rho_{\min}$ represents the lowest value of the evaporation coefficient.

At the beginning of the process, a relatively high evaporation coefficient is employed to promote extensive exploration and local search by the ants. Subsequently, this coefficient is gradually reduced, which extends the duration that pheromones persist on the paths. As a result, the impact of pheromones on the choice of paths is enhanced, aiding the algorithm in converging toward improved solutions.

F. Limiting the Pheromone Threshold

In order to enhance the solution quality obtained by the Ant Colony algorithm and maintain randomness during convergence, while effectively preventing premature convergence to local optima, the Max-Min Ant System strategy is adopted. Following each pheromone update, the pheromone levels on the updated nodes are constrained within specified bounds [14]. The principle governing the pheromone threshold limitation is described as follows:

$$\tau_{\text{ij}} = \begin{cases} \tau_{\min}, & \text{if } \tau_{\text{ij}} < \tau_{\min} \\ \tau_{\max}, & \text{if } \tau_{\text{ij}} > \tau_{\max} \\ \tau_{ji}, & \text{else} \end{cases} \tag{9}$$

By limiting the range of pheromone values to $[\tau_{\min}, \tau_{\max}]$, the pheromones of all paths found during the search are confined within this interval. This ensures that pheromones are concentrated on relatively optimal path nodes, which not only increases the likelihood of searching for other paths but also avoids excessive pheromone differences among path nodes that could cause the algorithm to prematurely converge.

IV. Adaptive ACO-Based Path-finding Algorithm

The adaptive pathfinding ant colony algorithm integrates three distinct mechanisms. Firstly, it introduces a guided direction mechanism to enhance the tendency of ants when selecting nodes [14]. Secondly, the initial pheromone update process is improved by employing the A* algorithm to generate a preliminary path, followed by an increased pheromone concentration around the grid cells along this route. Finally, the heuristic function of the ant colony algorithm incorporates the pathfinding strategy of the A* algorithm, allowing iterative path optimization to ultimately identify the optimal route. Through the fusion of these three mechanisms with the conventional ant colony algorithm, the adaptive pathfinding ant colony algorithm is formulated [15].

A. Guided Direction Mechanism

The core concept of the ant colony algorithm involves several steps: initially, pheromone concentrations on all possible paths are set to starting values. Next, ants probabilistically select their next nodes and build complete paths. Throughout this process, path lengths are recorded, and pheromone trails are updated accordingly. The shortest path discovered so far is preserved, while both pheromone levels and tabu lists are refreshed. This iterative cycle repeats until the stopping criterion is met. Ants determine their target nodes by considering pheromone intensity along with heuristic information. However, during the very first iteration, all ants must traverse the entire map to establish an initial rough path, which incurs considerable computational time.

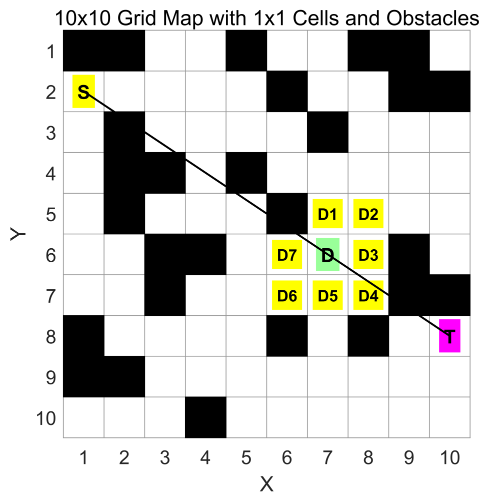


Fig. 4. Optional Nodes for the Original Ant Colony Algorithm.

As shown in Fig. 4 in the traditional ant colony algorithm, an individual moves toward the surrounding nodes. Around the current node D, there is an obstacle grid (black block). The selection of the next node requires calculating the transition probabilities of seven nodes (D1-D7), resulting in a longer computation time in the planning of the route [16].

To address the limitations of the traditional ant colony algorithm in node selection, a guiding direction mechanism is introduced to steer the choice of nodes, thereby effectively reducing computational complexity. This mechanism utilizes directional information from the start point S to the end

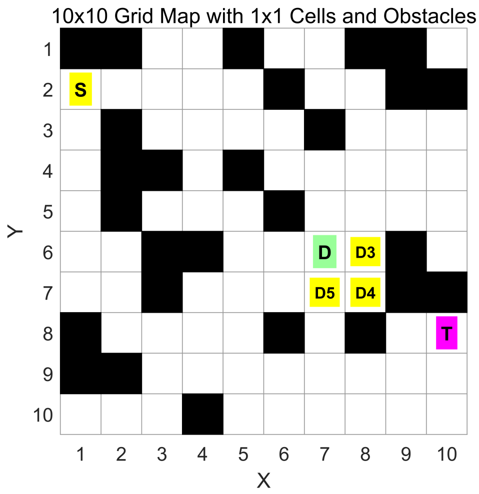


Fig. 5. Schematic Diagram of the Guidance Direction Mechanism.

point T to constrain the selection of the next node, cutting down the original set of seven possible nodes to only three, which enhances the efficiency of the path-finding process. As illustrated in Fig. 5, a coordinate system is established with the starting node S serving as the origin. The subsequent node chosen after the current node must align with the direction from S toward T. Let the coordinates of the current node D be (x_c, y_c), and those of a candidate next node D_n be (x_d, y_d). According to the guiding direction criteria, the difference in the x-coordinate $x_{d-c} = x_d - x_c$ should be greater than or equal to zero, while the difference in the y-coordinate $y_{d-c} = y_d - y_c$ must be less than or equal to zero. Consequently, among the seven candidate nodes D_1 through D_7 depicted in Fig. 4, only three nodes—D_3, D_4, and D_5—satisfy these conditions.

B. *Incorporating the Path-finding Mechanism of A* Algorithm into the Ant Colony Algorithm*

Furthermore, when obstacles occupy nodes D_3 and D_5 adjacent to the current node, the guiding direction mechanism fails to provide any feasible next nodes. In such situations, the node selection strategy switches to the pathfinding method of the A* algorithm. The A* algorithm is a deterministic search technique that merges the benefits of Dijkstra's algorithm and Breadth-First Search. It is widely recognized as one of the most efficient approaches for determining the shortest path length within grid-based models.

$$\begin{aligned} F(n) &= G(n) + H(n) \\ G(n) &= \sqrt{(x_n - x_s)^2 + (y_n - y_s)^2} \\ H(n) &= \sqrt{(x_n - x_e)^2 + (y_n - y_e)^2} \end{aligned} \tag{10}$$

In this context, G represents the movement cost from the start point S to the end point T along the generated path passing through the required grids. Meanwhile, H denotes the estimated cost from a given grid to the end point T. As illustrated in Fig. 6, when the current position is at node D_4, the guiding direction mechanism is no longer applicable, and the node selection strategy switches to the A* path-finding algorithm.

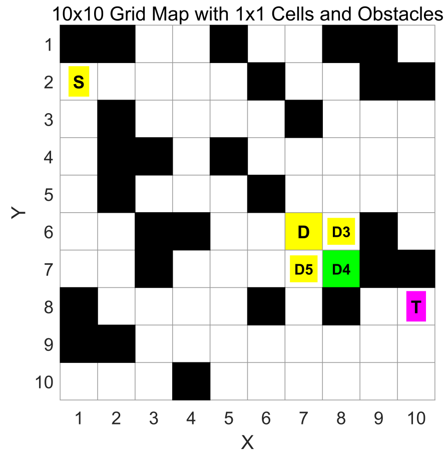


Fig. 6. Schematic Diagram of Path-finding.

In this node selection framework, the initial pheromone intensity is defined by

$$\tau_0 = \frac{m}{L_m},$$

where m represents the total number of ants, and L_m denotes the cost of the path from the current node to the target T, calculated as

$$L_m = G(n) + H(n).$$

Accordingly, the probability that an ant selects the subsequent node is formulated as:

$$p_{\mathrm{ij}}^{\mathrm{k}}(\mathrm{t}) = \begin{cases} \dfrac{[\tau_{\mathrm{ij}}(\mathrm{t})]^{\alpha}\,[\eta_{\mathrm{ij}}(\mathrm{t})]^{\beta}}{\sum_{\mathrm{i}\in \mathrm{a_k}} [\tau_{\mathrm{is}}(\mathrm{t})]^{\alpha}\,[\eta_{\mathrm{is}}(\mathrm{t})]^{\beta}}, & \mathrm{j} \in \mathrm{a_k} \\ 0, & \text{others} \end{cases} \tag{11}$$

Here, τ_0 denotes the pheromone level on the edge (i, j), η_{ij} represents the heuristic factor between nodes i and j, and α_k is the set of nodes that ant k is allowed to visit in the next step.

C. *Improving Initial Pheromone with A* Algorithm*

In the conventional ant colony algorithm, the initial pheromone concentration is uniformly assigned to all nodes on the map, which often leads to an extended duration for the first iteration [17]. To address this, the A* algorithm is utilized to enhance the initial pheromone distribution. Specifically, the A* algorithm is first applied to generate a suboptimal path, after which the pheromone concentration along this path is increased. This strategy of leveraging A* to improve the initial pheromone levels more effectively directs the ants during path searching and consequently speeds up the algorithm's convergence [18].

$$\begin{cases} \tau_{\mathrm{ij}}(\text{ initial }) = \dfrac{\mathrm{d_{min}}}{\mathrm{d_{SI}} + \mathrm{d_{iT}}}\mathrm{C}(0), & \text{freegrids} \\ \tau_{\mathrm{ij}}(\text{ initial }) = \dfrac{0}{\mathrm{d_{SI}} + \mathrm{d_{iT}}}, & \text{obstaclegrids} \end{cases} \tag{12}$$

Here, d_{SI} denotes the distance from the start point S to the current node i, while d_{iT} represents the distance from node i to the target T. Additionally, the value of $C(0)$ is set to 1, corresponding to the initial pheromone concentration used in the original ant colony algorithm.

Compared to the conventional ant colony algorithm, the proposed approach for enhancing the initial pheromone distribution allows individual ants to more effectively recognize the dominant network. As illustrated in Figure 7, the path d_{ST} represents an initial route planned by the A^* algorithm. According to the method described in this paper, the initial pheromone concentration assigned to nodes along d_{ST} reaches a maximum value of 1. Moreover, the closer a current node is to the path d_{ST}, the higher the initial pheromone value attributed to the corresponding grid. Consequently, this improved initialization technique enables ants to conduct path searching in a more focused and efficient manner compared to the traditional ant colony algorithm.

Moreover, to prevent the algorithm from becoming trapped in local optima, it is essential to impose upper and lower bounds on the pheromone concentration. Following the Max-Min Ant System (MMAS) approach, the pheromone concentration $\tau_{ij}(t)$ is constrained within the interval $[\tau_{\min}, \tau_{\max}]$ after each update, as expressed in Equation (13). Here, $\tau_{ij}(t)$ denotes the pheromone level on the edge between nodes i and j during the t-th iteration, while $\tau_{\max}$ and $\tau_{\min}$ represent the respective maximum and minimum pheromone limits.

$$\tau_{\min} \leq \tau_{\mathrm{ij}}(\mathrm{t}) \leq \tau_{\max} \tag{13}$$

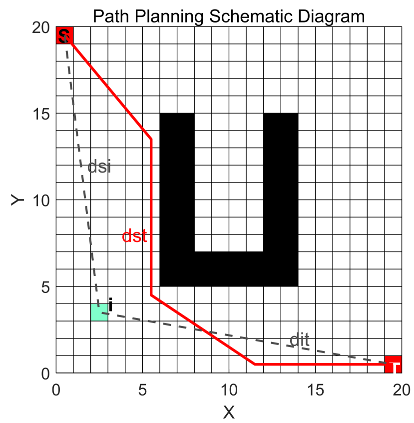


Fig. 7. Schematic Diagram of the Initial Pheromone.

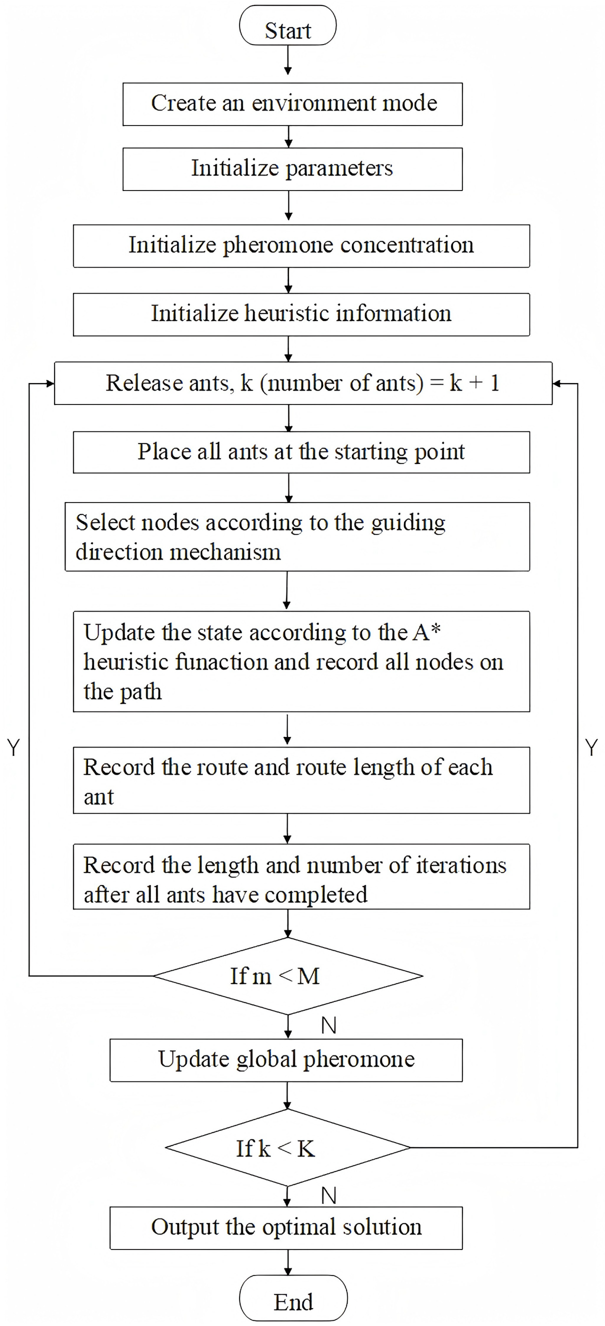


Fig. 8. Flowchart of the Improved ACO Algorithm.

Multiple prior studies have investigated the integration of A* with Ant Colony Optimization (ACO) to enhance path-planning effectiveness. For instance, Zhu et al. [7] introduced a variable-grid ACO algorithm that employs A* to guide the initialization of pheromones, thereby accelerating convergence during the early iterations. Similarly, Liu et al. [11] embedded heuristic information from A* into the ant's movement strategy within complex environments, resulting in improved path smoothness.

However, most of these approaches focus on static fusion strategies and lack adaptive control in response to environmental changes. In contrast, the algorithm proposed in this paper not only integrates the A* heuristic into the transition probability calculation and pheromone initialization, but also introduces a direction-guided mechanism and adaptive grid size adjustment, enabling dynamic responsiveness to obstacle density and search context. Furthermore, a reward-punishment pheromone update strategy with adaptive evaporation coefficient is incorporated to improve global search capability and reduce the risk of premature convergence - features not fully addressed in the existing literature.

D. Comparison with Existing Methods

Compared to existing methods that integrate A* with ACO [7, 11], this work offers a more comprehensive improvement framework. Unlike static pheromone initialization approaches, our method dynamically adapts grid size based on obstacle density and further introduces a directional guidance mechanism to reduce search complexity. In addition, the proposed adaptive pheromone evaporation rate and reward-punishment mechanism enhance global search ability and convergence robustness. These innovations jointly address the issues of premature convergence, redundant turns, and poor early-stage search quality that are often found in existing hybrid methods.

V. Flow of Adaptive Path-finding Ant Colony Algorithm

This work introduces three enhancement strategies compared to the traditional ant-based optimization method: incorporating a guiding direction mechanism, utilizing A* to initialize pheromone levels more effectively, and embedding the A* path-finding approach within the ant-inspired framework [19]. Building upon these improvements, an enhanced algorithm—referred to as the adaptive path algorithm—is proposed for robot path planning. The flowchart depicting this algorithm is shown in Fig. 8.

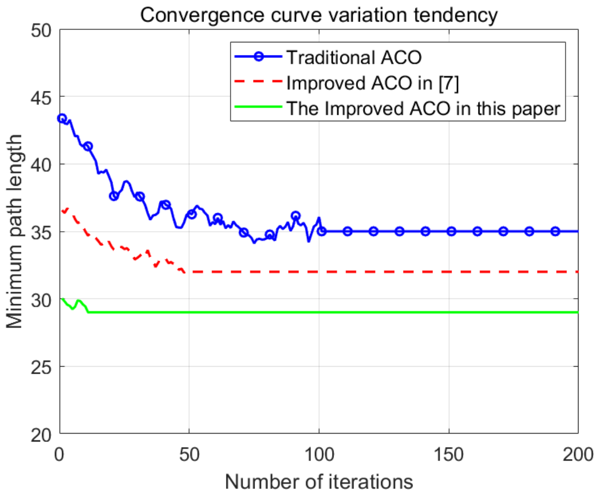


Fig. 9. Comparison of Iteration Trends.

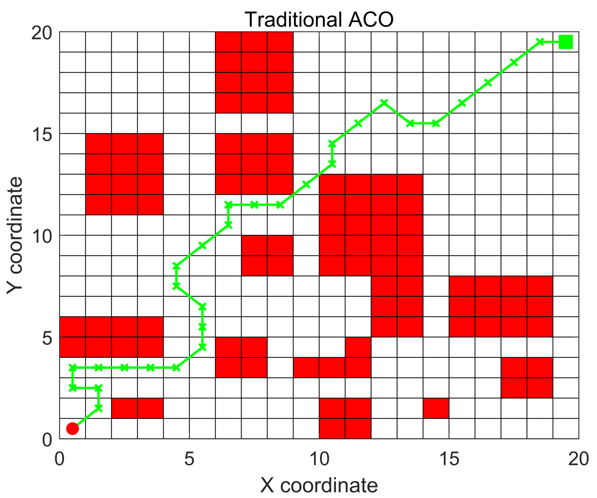


Fig. 10. Path Optimization Curve of Traditional Algorithm.

VI. Simulation Experiment and Structural Analysis

A. Experimental Setup

The simulation experiments were conducted using MATLAB 2023b. For each combination of parameters, 20 independent simulations were performed to determine the optimal configuration, as summarized in Table I. Subsequently, path-planning tests were carried out on the grid model depicted in the figure to evaluate the performance of the proposed Ant Colony Optimization algorithm.

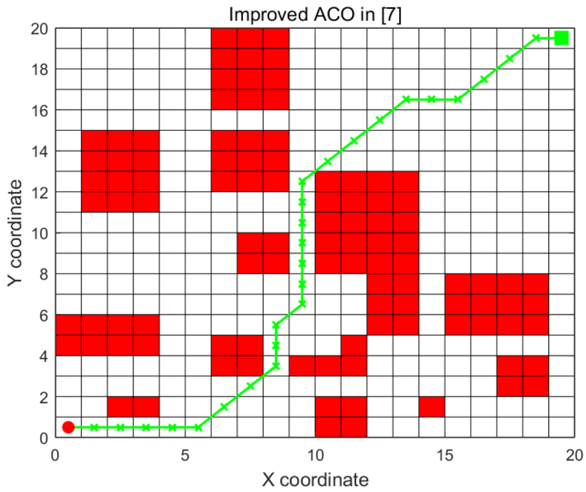


Fig. 11. Path Optimization Curve of the Improved ACO Algorithm in [7].

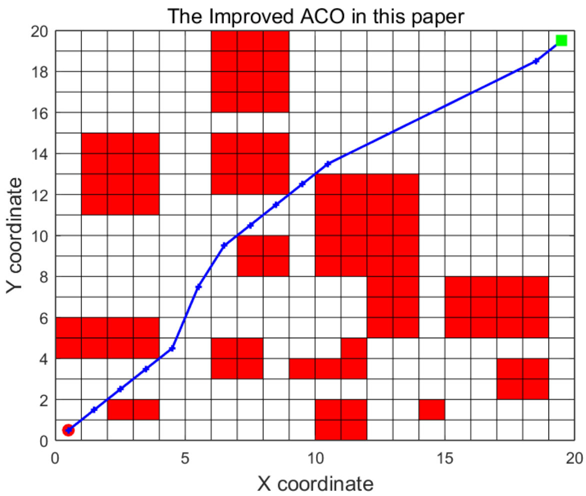


Fig. 12. Path Optimization Curve of Our Improved ACO Algorithm.

Figure 9 illustrates the convergence comparison among three ant colony optimization algorithms applied to path planning. The findings reveal that the proposed algorithm in this study achieves convergence around the 10th iteration, which is notably faster than both the traditional ant colony algorithm and the improved method presented in Reference [7]. Furthermore, the final optimal path length obtained is approximately 29.2, surpassing the results of the method in Reference [7] as well as the conventional algorithm. These experimental outcomes validate that the proposed enhancement mechanism significantly improves both the convergence rate and path quality, demonstrating superior optimization performance and greater practical applicability in engineering contexts.

Figures 10 to 12 compare the path-planning outcomes of the traditional ant colony algorithm, the improved version, and the ant colony algorithm augmented with variable grid technology within a 20×20 grid environment. The start point is positioned at grid coordinate (0, 0), while the target is located at (20, 20). The results indicate that the conventional algorithm tends to become trapped in local minima, resulting in suboptimal paths that are redundant and inefficient. In contrast, the improved algorithm with variable grid technology,

TABLE I
PARAMETER SETTINGS.

Element	Setting
Ant colony size m	100
Heuristic element β	6
Pheromone component α	1
Starting pheromone decay coefficient ρ	0.8
Lower bound of pheromone evaporation coefficient $\rho_{\min}$	0.1
Iterations T	100

designed for complex environments, effectively eliminates path redundancy, prevents ants from revisiting locally optimal areas, guides them around obstacles, and enables the discovery of the global optimal path.

Table II compares the path-finding durations of three ant colony-based algorithms. Among them, the variable-size grid ant colony algorithm achieves the fastest time of 0.1139 seconds, followed by the traditional ant colony algorithm with 0.2938 seconds, while the improved ant colony algorithm presented in [7] records the longest duration of 0.3447 seconds. These results demonstrate that the proposed variable-size grid ant colony algorithm exhibits the highest efficiency in pathfinding among the three methods.

TABLE II
COMPARISON OF IMPROVED ALGORITHMS.

Algorithm type	path-finding time/s
Traditional Ant Colony Algorithm	0.2938
Improved Ant Colony Algorithm in [7]	0.3447
ACO Method with Variable-Resolution Grid Structure	0.1139

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS.

Algorithm	Optimal Path Length	Average Convergence Iterations
Improved ACO	37.6	43
PSO	41.3	59
DE	39.8	71
GA	42.1	66

B. Comparative Analysis

To ensure a fair comparison, all algorithms were evaluated on the same map using identical start and goal positions, each running for 100 iterations. The genetic algorithm used a recombination probability of 0.8 and a mutation probability of 0.05; the particle swarm optimization algorithm utilized a population size of 30 with an inertia weight of 0.9; and the differential evolution algorithm applied a mutation factor of 0.5 alongside a crossover rate of 0.7. Each method was executed 20 times to compute average metrics including path length, number of iterations to convergence, and computation time.

Moreover, to thoroughly assess the performance of the enhanced ant colony optimization algorithm, this study conducts comparative experiments involving GA, DE, and PSO. These experiments use the same map, identical start and end points, and consistent operational parameters, gathering statistical measures such as path length, convergence iterations, and average computation time. As presented in Table III, the improved ACO algorithm surpasses the other three methods in both convergence speed and path quality, highlighting its superior robustness in dynamic obstacle environments.

VII. CONCLUSION

This study presents a robot path planning approach founded on an enhanced Ant Colony Optimization algorithm. To overcome the drawbacks of low initial search efficiency and vulnerability to local optima inherent in traditional ACO methods, the following improvements have been introduced:

- A differentiated pheromone distribution strategy was implemented to optimize the initial pheromone allocation, enhancing the ant colony's search efficiency and improving path planning performance.
- A dynamic pheromone updating strategy combining a reward-punishment mechanism and an adaptive evaporation coefficient was adopted to increase the global search capacity and avoid local optima.

The experimental findings indicate that the enhanced ACO algorithm achieves superior global search performance, faster convergence, and improved path quality in a variety of complex map environments.The generated paths are not only shorter and smoother but also display better stability and robustness in repeated experiments.

REFERENCES

[1] N. Anil, Y. Sukhi, M. Preetha, and K. Sivakumar, "Ant colony optimization with levy-based unequal clustering and routing (aco-ucr) technique for wireless sensor networks," *Journal of Circuits, Systems and Computers*, vol. 33, no. 03, p. 2450043, 2024.

[2] B. Guan, Y. Zhao, and Y. Li, "An improved ant colony optimization with an automatic updating mechanism for constraint satisfaction problems," *Expert Systems with Applications*, vol. 164, p. 114021, 2021.

[3] I. Kale, M. Sapre, A. Khedkar, K. Dhamankar, A. Anand, and A. Singh, "Hybrid aco-ci algorithm for beam design problems," *SN Computer Science*, vol. 5, no. 3, p. 282, 2024.

[4] H. Tian, "Research on robot optimal path planning method based on improved ant colony algorithm," *International Journal of Computing Science and Mathematics*, vol. 13, no. 1, pp. 80–92, 2021.

[5] K. Athira, R. Yalavarthi, T. Saisandeep, K. Harshith, A. Sha, and D. Udayan, "Aco-dtsp algorithm: optimizing uav swarm routes with workload constraints," *Procedia Computer Science*, vol. 235, pp. 163–172, 2024.

[6] X. Li and L. Wang, "Application of improved ant colony optimization in mobile robot trajectory planning," *Mathematical Biosciences and Engineering*, vol. 17, no. 6, pp. 6756–6774, 2020.

[7] Z. Liu, G. Nan, Y. Cheng, H. Du, and W. Zhu, "A transmission line planning method based on variable size grid and improved ant colony optimization algorithm," *2023 38th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 838–843, 2023.

[8] J. Revanna and N. Al-Nakash, "Impact of aco intelligent vehicle real-time software in finding shortest path," *Software Impacts*, vol. 19, p. 100625, 2024.

[9] J. Si and X. Bao, "A novel parallel ant colony optimization algorithm for mobile robot path planning," *Mathematical Biosciences and Engineering*, vol. 21, no. 2, pp. 2568–2586, 2024.

[10] M. Shi, S. Xiao, and X. Feng, "An adaptive ant colony algorithm based on local information entropy to solve distributed constraint optimization problems," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 37, no. 09, p. 2359013, 2023.

[11] W. Zhu, W. Cai, and H. Kong, "Optimal path planning based on aco in intelligent transportation," *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 441–450, 2025.

[12] W. Fang, Z. Liao, and Y. Bai, "Improved aco algorithm fused with improved q-learning algorithm for bessel curve global path planning of search and rescue robots," *Robotics and Autonomous Systems*, vol. 182, p. 104822, 2024.

[13] B. Yan, J. Quan, and W. Yan, "Three-dimensional obstacle avoidance harvesting path planning method for apple-harvesting robot based on improved ant colony algorithm," *Agriculture*, vol. 14, no. 8, p. 1336, 2024.

[14] W. Gao, Q. Tang, B. Ye, Y. Yang, and J. Yao, "An enhanced heuristic ant colony optimization for mobile robot path planning," *Soft Computing*, vol. 24, pp. 6139–6150, 2020.

[15] H. Li, X. Zhang, S. Fu, and Y. Hu, “A hybrid algorithm based on ant colony optimization and differential evolution for vehicle routing problem,” *Engineering Letters*, vol. 29, no. 3, pp. 1201–1211, 2021.
[16] Y. Hou, B. Liu, L. Dang, W. He, and W. Gu, “A local search-based metaheuristic algorithm framework for the school bus routing problem,” *Engineering Letters*, vol. 30, no. 1, pp. 17–28, 2022.
[17] F. Ni, B. Jin, X. Gao, and Q. Li, “Cooperative obstacle avoidance planning via a*-bdwa fusion for unmanned aerial vehicles,” *IAENG International Journal of Computer Science*, vol. 51, no. 11, pp. 1793–1803, 2024.
[18] Y. Luo, X. Zhang, and J. Wu, “An improved hybrid particle swarm optimization path planning algorithm based on particle reactivation,” *IAENG International Journal of Computer Science*, vol. 51, no. 10, pp. 1534–1545, 2024.
[19] P. Kusuma, “Best couple algorithm: a new metaheuristic with two types of equal size swarm splits,” *IAENG International Journal of Applied Mathematics*, vol. 54, no. 8, pp. 1615–1623, 2024.