Mathematical Modeling and Algorithm Application
ISSN: 3006-0842 | Vol. 7, No. 3, 2025

Path Planning for Mobile Robots Based on Ant Colony Algorithm Integrated with Time Warping Algorithm

Junqi Wu*, Bo Huang, Chongshan Zhou

School of Mechanical Engineering, Sichuan University of Science and Engineering, Yibin 644000, China
* **Corresponding author**: Junqi Wu

Abstract: To address the slow convergence and tendency to fall into local optima of traditional ant colony algorithms in complex indoor environments, a new method integrating ant colony algorithm and time warping algorithm is proposed. This method improves convergence speed by adding a dynamic decay function to the heuristic function; prevents the algorithm from getting stuck in local optima by introducing a random disturbance function to dynamically adjust pheromones; enhances exploration capability by using the Dynamic Time Warping (DTW) algorithm to evaluate path similarity and assign "novelty" to planned paths; improves global search ability by introducing crossover operations from genetic algorithms; and finally smooths the output path using B-spline curves. Simulation in MATLAB shows that the GACOMR algorithm converges in 7 iterations with an optimal path length of 58.84m, outperforming other algorithms. Physical experiments with a mobile robot further confirm its feasibility.

Keywords: Ant Colony Algorithm; Path Planning; Heuristic Function; Pheromone Improvement; B-Spline Curve

1. Introduction

With the rapid advancement of robotics technology both domestically and internationally, mobile robots have gradually gained widespread application in various critical fields [1]. For instance, they can replace manual labor in transporting goods to designated locations [2], be used for agricultural seeding, and operate under harsh environmental conditions [3]. Path planning provides mobile robots with relatively optimal operation routes [4] and helps avoid inefficient paths. Path planning algorithms primarily include traditional algorithms and intelligent algorithms. Traditional algorithms mainly consist of the A* algorithm [5], Dijkstra's algorithm [6], and the artificial potential field method [7]. Intelligent algorithms mainly include genetic algorithms [8], ant colony algorithms [9], and particle swarm optimization [10].

The traditional ant colony algorithm, which simulates the collective foraging behavior of ants, has garnered significant attention [11]. Consequently, scholars worldwide have conducted extensive research on it. Regarding the heuristic function, Reference [12] enhanced the guidance of the ant colony in path selection by introducing the target direction distance and amplifying it. An adaptive factor was also incorporated to prevent the algorithm from converging to local optima. Reference [13] improved the algorithm's optimization capability by incorporating the relationship between the current grid node and the target grid node and amplifying the distance difference. However, the weighting factor in such methods remains constant, which can significantly diminish their guiding effectiveness for the ant colony in complex environments. Concerning pheromone updating, Reference [14] proposed a pheromone update method based on entropy weight, enabling faster convergence of the ant colony. By adopting a segmented pheromone update approach, path diversity was increased. Reference [15] accelerated convergence by incorporating a turning angle penalty factor and a distance factor to establish a pheromone reward-punishment mechanism. Reference [16] utilized auxiliary ant colonies combined with their directional advantages to assign initial pheromones for the main ant colony. Such methods may cause ants to concentrate on relatively optimal paths while overlooking other excellent ones. Regarding initial path research, Reference [17] used the A* algorithm to plan an initial path to prevent ants from blindly searching during the initial stages. Reference [18] set pheromone trail quantities within a specific interval to avoid blind initial searches. However, initial path guidance strategies rely on specific paths or map structures, which can easily lead to unidirectional searches and insufficient path diversity.

Although the aforementioned scholars have improved the traditional ant colony algorithm, it may still encounter local optima when facing U-shaped traps or narrow passages. This indicates that there remains room for further improvement in the ant colony algorithm. To address these issues, the GACOMR algorithm introduces an adaptive adjustment function to enhance the heuristic function, reducing the dominance of the heuristic factor and increasing the likelihood of escaping local optima. This enables the algorithm to avoid being trapped near obstacles due to the proximity of shorter routes. A random disturbance factor is introduced to effectively balance diversity exploration in the early stages and stable convergence in the later stages. The sequence crossover operation from genetic algorithms is incorporated to optimize global paths, preventing the ant colony from getting stuck in local optima. Finally, B-spline curves are employed to smooth the planned path. The improved algorithm can generate a smooth and globally optimal path, ensuring that AGVs can achieve autonomous optimization in complex environments.

2. Ant Colony Algorithm

2.1. Map Environment Construction

Environment modeling is the first step in path planning for mobile robots. Common environment modeling methods include the grid method, visibility graph method, and

topological graph method. The grid method offers advantages such as easy construction, simple processing, and effective visualization. Therefore, the grid method is selected for environment modeling.

The grid method planarizes the environment and performs binary representation of all objects: obstacles are denoted as 1, represented by black grids, indicating non-traversable areas. Non-obstacles are denoted as 0, represented by white grids, indicating traversable areas. The grid map is shown in Figure 1.

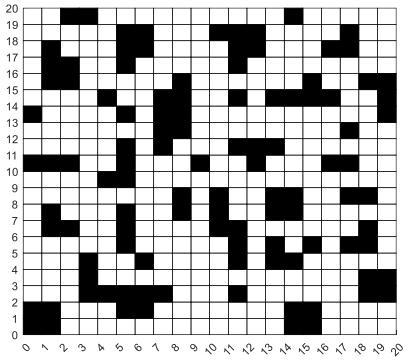


Figure 1. Raster map construction

2.2. Traditional Ant Colony Algorithm

The Ant Colony Algorithm is inspired by the foraging behavior of ants in nature. Ants depart from their nest and leave pheromone trails while searching for food to guide subsequent ants. As the number of ants increases, the pheromone concentration along shorter paths to the food source becomes stronger, attracting even more ants.

2.2.1. Path Selection

The grid method models the mobile robot and the experimental environment in two dimensions. The mobile robot starts from the initial point, with each grid initially having the same pheromone level. The probability of the mobile robot selecting the next grid point is given by Equation 1, which is determined by both heuristic information and pheromone levels.

$$p_{ij}^{k}=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s\in C_k}\tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)}, s\in C_k\\ 0, s\notin C_k\end{cases} \tag{1}$$

In the equation: α is the pheromone heuristic factor, which controls the degree to which ants rely on pheromones.β represents the influence of heuristic information on the path selection of ants, guiding their direction. τ_{ij} denotes the pheromone concentration from node i to node j. η_{ij} represents the heuristic function from node i to node j.The heuristic function provides initial heuristic information to the ants, helping the ant colony choose paths that are closer to the optimal solution. The heuristic function is defined as shown in Equation 2.

$$\eta_{ij}=\frac{1}{d_{ij}} \tag{2}$$

2.2.2. Pheromone Concentration Update

Each ant leaves a unique pheromone trail on any node it traverses. However, the pheromone deposited by ants gradually dissipates over time, while new ants continue to leave fresh pheromone trails. When an ant completes a path from the start to the goal, the pheromone on that path is updated accordingly. Paths of higher quality receive positive feedback, leading to an increase in pheromone concentration, making such paths more attractive to subsequent ants. The update rule is defined as shown in Equation 3.

$$\tau_{ij}(t+1)=(1-p)\cdot\tau_{ij}(t)+\Delta\tau_{ij} \tag{3}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{n}\Delta\tau_{ij} \tag{4}$$

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}\dfrac{Q}{L_k}\\ 0\end{cases} \tag{5}$$

In the equation: $\tau_{ij}(t+1)$ represents the pheromone concentration on the path from node i to node j after being updated upon an ant completing its traversal; $\tau_{ij}(t)$ denotes the pheromone persistence factor, indicating the extent to which pheromones from the current moment are retained; $\Delta\tau_{ij}$ signifies the total pheromone increment on the path from node i to node j after all ants complete their traversal; $\Delta\tau_{ij}^{k}$ refers to the pheromone increment contributed by the k-th ant on the path from node i to node j; L_k indicates the distance traveled by the k-th ant on the path from node i to node j; and Q is a constant representing the total amount of pheromone added each time an ant passes through a path.

3. Improved Ant Colony Algorithm

3.1. Improve heuristic function

The ant colony relies on heuristic functions to select optimal paths, indicating the crucial guiding role of heuristic functions in finding the best route. However, the traditional ant colony algorithm only considers the current grid node i and the candidate grid node j as guiding information, neglecting the directional influence of the target grid node g. This oversight often leads to aimless exploration of the map by the ant colony, resulting in slow convergence. To address this limitation, Reference [19] incorporated both the next grid node and the target grid node as guiding factors and introduced directional selection heuristics, thereby accelerating convergence. Nevertheless, the weight factor in the aforementioned method remains static, which significantly diminishes its guiding effectiveness in complex environments, leading to extensive inefficient searches and reduced efficiency.

To overcome these issues, this paper introduces a dynamic decay factor. The ratio of the current optimal path length to the global optimal path length is incorporated into the heuristic function as a weighting proportion, while the distance between the current node and the target node is added to enhance directional guidance. In the early stages of the algorithm, this approach reduces the dominance of the heuristic factor, allowing the ant colony to rely more on pheromone distribution for exploration and increasing the probability of escaping local optima. In the later stages, the guiding role of the heuristic factor is restored to expedite

convergence. The improved formula is as follows:

$$n_{ij}(t)=\frac{1}{d_{ij}+d_{jg}}(1-\mu(t)\cdot(\frac{T}{T_{\max}})^{q}) \tag{6}$$

In the equation, d_{jg} represents the distance from the next grid node j to the target grid node g, and $\mu(t)$ denotes the dynamic decay factor, which controls the decay rate of heuristic information in the algorithm, where T indicates the current exploration round. $T_{\max}$ represents the maximum exploration round. q is the decay exponent, which controls the decay rate of heuristic information. In the early stages of the algorithm, when 0<q<1, the decay is slower, allowing heuristic information to exert greater influence. In the later stages, when q>1, the decay accelerates, making the algorithm rely more on pheromones to speed up convergence. d_y / d_d represents the gap between the current optimal solution and the historical optimal solution. When the current optimal solution is close to the historical optimal solution, the role of the heuristic factor is reduced, and the algorithm relies more on pheromones to accelerate convergence. Conversely, if the current optimal solution deviates significantly from the historical optimal solution, the heuristic factor dominates, maintaining stronger exploratory behavior to avoid premature convergence.

$$\mu(t)=\mu_0\cdot(1-\frac{d_y}{d_d}) \tag{7}$$

In the equation, μ_0 represents the initial decay factor, d_d Current denotes the current optimal path length, and d_y indicates the global optimal path length.

3.2. Optimization of state transition rules

When selecting the next node j, the traditional ant colony algorithm uses a roulette wheel method to choose from the neighboring nodes of the current grid node i. However, this approach tends to prioritize exploring nodes with higher pheromone levels, making the algorithm prone to falling into local optima. To address this issue, this paper introduces a random disturbance factor to enhance the random exploration capability of the ant colony and improve its global optimization ability. The improved rule is as follows:

$$P_{ij}=\frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)\cdot(1+p(t)\cdot\Delta ij)}{\sum_{s\in C_k}\tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)\cdot(1+p(t)\cdot\Delta is)} \tag{8}$$

In the equation, $p(t)$ represents the adaptive adjustment factor for the disturbance factor, which can be adaptively tuned based on the iteration of the ant colony algorithm, while Δij and Δik denote the disturbance factors from grid node i to grid node j and from grid node i to grid node s, respectively, with each disturbance factor taking a random value in the range (0, 1). During the early stages of the ant colony algorithm, when the current iteration round T is low, the algorithm exhibits strong random disturbance, enhancing the ant colony's exploratory ability, and the selection of the next grid node does not overly rely on pheromone concentration. This not only strengthens the random exploration capability of the ant colony but also prevents premature convergence to local optima. As the algorithm progresses to later stages, the disturbance factor gradually increases from 0, reintroducing a controlled level of randomness as convergence advances, thereby maintaining moderate stochasticity and avoiding stagnation in local optima.

$$p(t)=\begin{cases}p_0\cdot(1-\frac{T}{T_{mid}}), T>T_{\max}\\ p_0\cdot(\frac{T-T_{mid}}{T_{\max}-T_{mid}}), T<T_{\max}\end{cases} \tag{9}$$

In the equation, T_{mid} is the adjustment parameter, and p_0 represents the magnitude of the initial disturbance factor, which controls the intensity of the disturbance factor in the early and late stages. When T approaches T_{mid}, the value of $p(t)$ is larger, indicating that the disturbance factor occupies a higher proportion in path selection, which favors diversified exploration. As T continues to increase, $p(t)$ gradually decreases, making path selection more inclined toward stable paths. The linear variation of $p(t)$ enables smooth changes in disturbance intensity, avoiding abrupt shifts and making it easier to control.

3.3. Optimization of pheromone update rules

The traditional ant colony pheromone update rule involves identifying ants that reach the target point after completing an iteration and increasing the pheromone concentration along their paths. However, this approach can lead to an excessive concentration of ants on suboptimal paths, influencing subsequent ant choices and resulting in higher pheromone levels on non-optimal paths. To address this issue, this paper employs an adaptive mechanism to dynamically adjust the pheromone contribution of each ant. Additionally, the Dynamic Time Warping (DTW) algorithm is introduced to evaluate path similarity. If a path exhibits significant differences from others, it is considered "novel," and the corresponding ant is granted the privilege of depositing more pheromone. In the early stages of the algorithm, the ant colony is assigned a larger pheromone increment, encouraging extensive exploration of path information, enhancing random exploration capabilities, and facilitating the discovery of more promising paths. In the later stages, the pheromone increment is reduced, allowing the ant colony to focus more on exploring paths near the optimal one, thereby accelerating convergence and minimizing ineffective searches. The improved formula is as follows:

$$\begin{cases}\tau_{ij}(t+1)=(1-p)\tau_{ij}(t)+\Delta\tau_{ij}^{k}(t)\\ \Delta\tau_{ij}^{k}(t)=Q[(1-\omega(t))\cdot s_k+\omega(t)\cdot\frac{1}{L_k}\end{cases} \tag{10}$$

In the equation, $\omega(t)$ represents the adaptive factor, which controls the pheromone intensity in the early and later stages of the algorithm, while S_k denotes the similarity degree.

$$\omega(t)=\frac{T}{T_{\max}} \tag{11}$$

$$S_k=1-\exp(-a\cdot D(p,q)) \tag{12}$$

In the equation, $D(p,q)$ represents the Dynamic Time

Warping (DTW) distance between the current path and the historical optimal path. When the DTW distance is small, it indicates minimal difference between the current path and the historical optimal path. Conversely, a larger DTW distance signifies greater dissimilarity, suggesting that the current path possesses novelty and thus qualifies for a higher pheromone allocation.

$$D(p,q) = D(m,n) \tag{13}$$

$$D(i,j) = d(p_i,q_j) + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases} \tag{14}$$

$$d(p_i,q_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{15}$$

In the equation, $i \in [1,m]$ and $j \in [1,n]$ represent the distances from the last element of the current path to the last element of the historical optimal path, where m and n respectively denote the final elements of the current path and the historical optimal path.

3.4. Introduce crossover operation

In the traditional ant colony algorithm, ants continuously search for paths while pheromones are dynamically updated. However, after multiple iterations, if no better solution is found, the algorithm tends to fall into a local optimum. To address this issue, a crossover operation is introduced to enable the ant colony to identify the global optimal solution. Since single-point or multi-point crossover methods, which involve random cuts and recombination, may lead to loops or unreachable nodes, sequential crossover is adopted in this study to maximize the preservation of the original path order and resolve node duplication issues after crossover. Specifically, two crossover points are used, triggered when no better solution is found after multiple consecutive iterations. The gene segment between the two crossover points from parent X is fixed in offspring x, while the remaining gene segments are sequentially filled from the other parent Y into offspring y. $l_1 = 2$, $l_2 = 4$.

Parent X = [1, 2, 3, 4, 5, 6, 7]

Parent Y = [1, 5, 2, 7, 6, 4, 3]

Extract the segment [2, 3, 4] from parent X and place it in the corresponding positions of the offspring.

Offspring = [-, 2, 3, 4, -, -, -]

Remove the genes in parent Y that are already present in the offspring segment.

Parent Y (remaining) = [1, 5, 7, 6]

Fill the remaining positions in the offspring sequentially with the genes from the modified parent Y.

Updated offspring = [1, 2, 3, 4, 5, 7, 6]

3.5. Path smoothing

Since the ant colony algorithm in grid-based path planning can only move from the current grid node to neighboring candidate nodes within the nine surrounding cells, this often results in numerous turning points, leading to a path that is not smooth enough for the robot and causing unstable operation. Therefore, path smoothing is essential. Given that B-spline curves are continuous and differentiable up to the second and third orders, and do not affect the output of the globally optimal path, the GACOMR algorithm employs B-spline smoothing curves to smooth the output path. The mathematical expression of the B-spline curve is as follows:

$$P(x) = \sum_{i=0}^{u} F_{i,n}(x) H_i \tag{16}$$

In the equation, $P(x)$ represents the B-spline curve, $F_{i,n}(x)$ denotes the basis function of the B-spline curve, and H_i refers to the control points on the curve.When the order of the B-spline curve is 3, the formula is as follows:

$$P(x) = TMH \tag{17}$$

$$T = [1 \quad x \quad x^2 \quad x^3] \tag{18}$$

$$M = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \tag{19}$$

$$H = \begin{bmatrix} H_0 \\ H_1 \\ H_2 \\ H_3 \end{bmatrix} \tag{20}$$

In the equation, T represents the parameter vector, M denotes the B-spline basis function coefficient matrix, and H refers to the control points on the curve.If smoothing is performed without constraints, the path may intersect obstacles, reducing its feasibility. To address this, collision detection is incorporated before smoothing, and a segmented smoothing operation is applied to constrain the smoothing process. The path length before smoothing is 29.67 m, and after smoothing, it is reduced to 28.45 m, representing a 4% improvement in length.

3.6. Algorithm Improvement Process

The algorithm flowchart is shown in Figure 2:

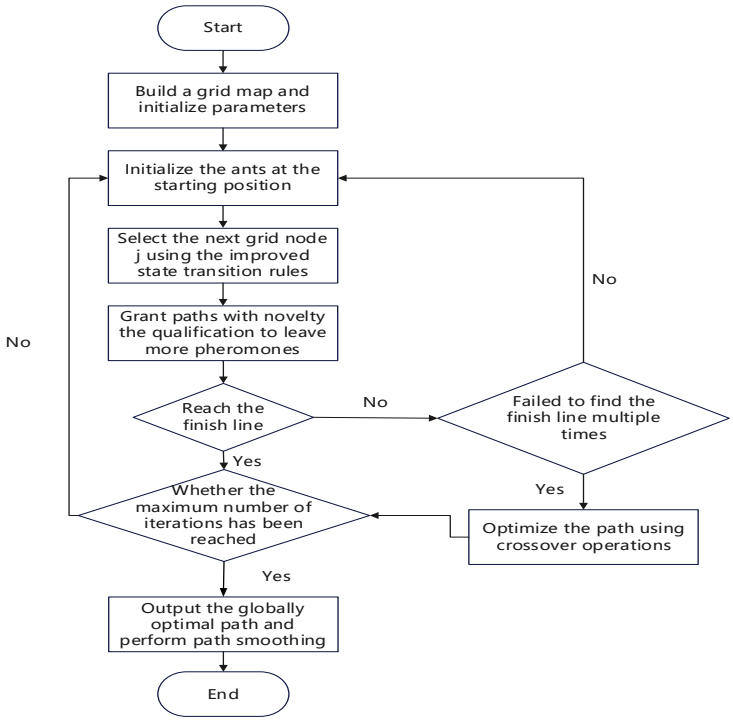


Figure 2 Algorithm flow

Algorithm Steps:

1.Construct the map using the grid method and initialize all parameters in the algorithm.

2.Place the ants at the starting point of the grid and begin the path search.

3.Use the improved state transition rule to select the next grid point, incorporating random disturbances to enhance the exploratory behavior of the ants.

4.Apply the improved pheromone update rule, increasing pheromone levels for paths with novelty to avoid local optima.

5.Check whether an ant has successfully found a complete path. If yes, proceed to Step 6; otherwise, return to Step 2.

6.Introduce a crossover operation. If the target is not reached after multiple attempts, perform crossover and recombination between the current optimal path and the historical optimal path to update the path.

7.Determine whether the maximum number of iterations has been reached. If yes, proceed to Step 8; otherwise, return to Step 2.

8.Smooth the output optimal path using the specified method, output the processed path, and terminate the algorithm.

4. Simulation and Experiment

To verify the optimization capability of the GACOMR algorithm in various complex environments, experiments were conducted using 20×20 and 40×40 complex grid environments with a large number of randomly shaped obstacles added to replicate the complexity of real-world scenarios. All experiments were validated in MATLAB R2023a. The parameters of the ant colony algorithm, the algorithm from Reference [20], and the improved GACOMR ant colony algorithm are detailed below.

Table 1. The improved GACOMR

Name	Numerical value	Name	Numerical value
m	50	$N_{\max}$	100
α	1	β	7
ρ	0.7	μ_0	0.5
p_0	0.2	T_{mid}	50

4.1. 20×20 Simulation Environment

In a 20×20 grid environment, MATLAB was used to simulate the improved mobile robot path planning based on the GACOMR algorithm, comparing it with the basic ant colony algorithm and the improved ant colony algorithm from Reference [20]. Each algorithm was run 20 times to verify the performance of the GACOMR algorithm. The starting point for the ants was set at (0.5, 19.5), and the endpoint was set at (19.5, 0.5). Figure 3 shows the optimal paths and iteration counts of each algorithm, while the final experimental results are compared in Table 2.

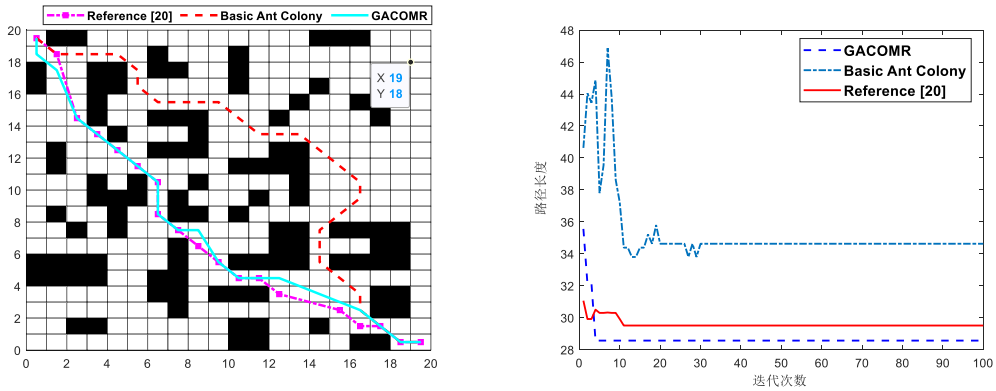


Figure 3. 20×20 simple raster environment

Table 2. Simulation Environment of 20×20

Environment	Algorithm	Path length / m			Number of iterations		Number of turns	
		Optimal	The worst	average	Optimal	average	Optimal	average
20×20	Ant Colony	34.62	38.04	36.47	30.00	34.25	15	16.95
	Reference [20]	29.50	30.54	30.12	11.00	13.80	11	12.65
	GACOMR	28.33	28.43	28.35	4.00	4.65	11	12.05

As shown in Figure 3 and Figure 4, compared to the optimal path of the traditional ant colony algorithm, both the method from Reference [20] and the GACOMR algorithm exhibit paths significantly closer to the optimal route. The optimal path of the GACOMR algorithm is slightly superior to that of Reference [20] and shows substantial improvement over the basic ant colony algorithm. From Table 3, it can be observed that the basic ant colony algorithm is prone to falling into local optima, requiring 30 iterations to converge to the optimal path, while the method from Reference [20] requires only 11 iterations to find the optimal path. Through improvements, the GACOMR algorithm requires only 4 iterations to locate the optimal path, demonstrating clear superiority over the other two algorithms. This indicates that the GACOMR algorithm is better suited for AGV path planning and meets practical requirements.

4.2. 40×40 Simulation Environment

To verify the performance of the GACOMR algorithm under complex conditions, simulations were conducted in a 40×40 grid environment using MATLAB. The number of ants in the public parameters was increased to 80, while all other parameters remained unchanged. Each algorithm was run 20 times, and the simulation results in the 40×40 grid environment are shown in Figure 5.

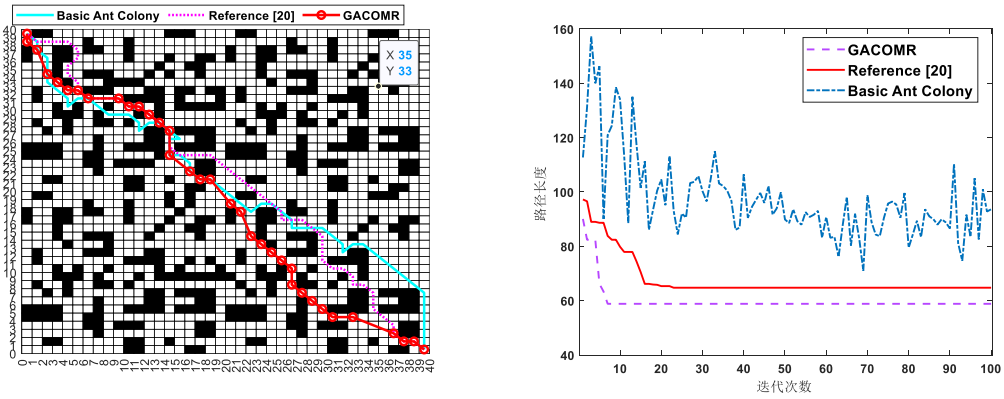


Figure 5. 40×40 complex environment

Table 3. Experimental results of 40×40

Environment	Algorithm	Path length / m			Number of iterations		Number of turns	
		Optimal	The worst	average	Optimal	average	Optimal	average
40×40	Ant Colony	74.66	78.08	76.46	-	-	33	37.55
	Reference [20]	64.76	66.18	65.53	23	24.75	32	34.20
	GACOMR	58.84	59.06	58.93	7	7.36	22	22.95

From the simulation results in each figure, it can be observed that the traditional ant colony algorithm performs poorly in finding the optimal path in complex environments. As shown in Figure 5, the traditional ant colony algorithm is easily affected by U-shaped obstacles in complex environments, exhibits a high number of turns, produces

longer optimal paths, and tends to fall into local optima. The optimal path values fluctuate without converging to the global optimum, indicating that the heuristic guidance of the traditional ant colony algorithm is inadequate. Reference [20] introduces the angle between the target node, the next candidate node, and the current node into the heuristic function, enhancing the guidance of the heuristic function in the early stages of the algorithm and improving convergence speed. This approach achieves convergence in only 23 iterations and significantly improves the optimal path length to 64.76 m. However, due to the high complexity of the map, the algorithm's iteration count and path distance do not reach the optimal level. After further improvements, the GACOMR algorithm achieves an optimal path length of 58.84 m and converges in only 7 iterations, outperforming both the other two algorithms. Compared to the 20×20 grid map, the 40×40 grid map is significantly more complex, featuring numerous U-shaped obstacles and various irregular obstacles. Nevertheless, the GACOMR algorithm maintains strong performance, demonstrating its excellent adaptability and effectiveness in diverse complex environments.

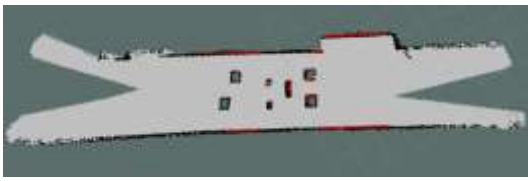
4.3. AGV Experimental Verification

In the previous chapter, simulations of the GACOMR algorithm were conducted in both simple and complex grid environments, verifying that the algorithm maintains strong performance even in complex settings. To further validate the GACOMR algorithm in real-world conditions, tests were performed using an AGV (Automated Guided Vehicle), as shown in Figure 6. The physical AGV is equipped with multiple modules, including LIDAR, cameras, and an IMU. To evaluate the performance of the GACOMR algorithm, a real-world experimental environment was constructed, as illustrated in Figure 7.

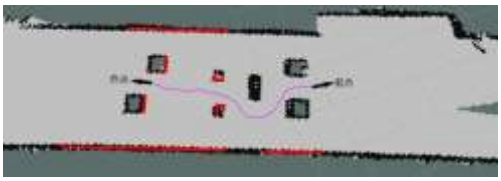
Figure 6. AGV trolley

Figure 7. Real-world site

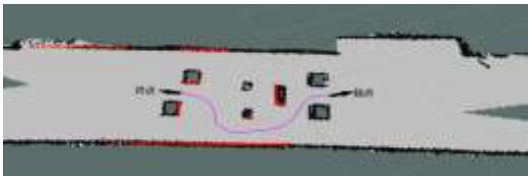
The AGV was placed at the starting point of the map, with various obstacles arranged in the environment. The Gmapping algorithm was used to construct a map of the actual environment through the rviz visualization tool. The GACOMR algorithm was then employed to plan a globally optimal path, which was compared with paths generated by the RRT algorithm and the ACO algorithm. The feasibility of the GACOMR algorithm in real-world scenarios was ultimately validated. A comparison among the ACO algorithm, RRT algorithm, and GACOMR algorithm is illustrated in Figure 8. The evaluation metric used was path length. The path length generated by the ACO algorithm was 7.51 m, by the RRT algorithm was 7.32 m, and by the GACOMR algorithm was 6.92 m. It can be observed that the ACO algorithm produced the longest path, while the GACOMR algorithm yielded the shortest path. Compared to the ACO algorithm, the GACOMR algorithm achieved a 7.85% improvement, and compared to the RRT algorithm, a 5.46% improvement. These results demonstrate the feasibility of the GACOMR algorithm in real-world environments.



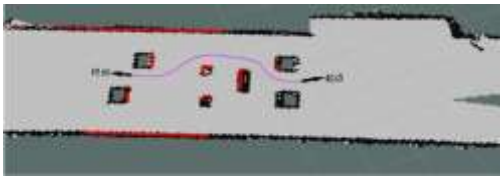
(a) Real-world environment mapping



(b)RRT algorithm



(c)ACO algorithm



(d)GACOMR algorithm

Figure 8. Experimental results

5. Summary

5.1. Summary

To address the issues of traditional ant colony algorithms in complex environments (especially those with various U-shaped traps), such as susceptibility to local optima, slow convergence, and blind searching, this paper proposes multi-faceted improvement strategies and conducts simulations as well as physical validations.

The main improvements of the GACOMR algorithm include the following aspects:

1.Introducing the ratio of the global optimal path length to the current optimal path length as a dynamic decay factor into the heuristic function. This allows the algorithm to rely more on heuristic information in the early stages for higher pathfinding efficiency, while reducing dependence in later stages to enhance random search capability, effectively improving convergence speed.

2.Incorporating a random disturbance factor into the state transition rules to increase the exploratory randomness of ants and prevent premature convergence to local optima.

3.Innovatively using the dynamic time warping (DTW) distance between the current path and the historical optimal path to evaluate path novelty during the pheromone update phase. Paths with higher novelty are assigned more pheromones, enhancing the algorithm's diversified search capability.

4.Introducing sequential crossover operations from genetic algorithms to recombine the current optimal path and the historical optimal path, further improving global search capability.

5.Employing B-spline curves to smooth the output path, thereby enhancing path feasibility.

Simulation results in MATLAB show that in a 20×20 simple grid environment, the GACOMR algorithm achieves an optimal path length of 28.33 m, approximately 18.1% shorter than the 34.62 m of the traditional ant colony algorithm. The convergence iteration count is reduced from 30 to 4, a decrease of 86.7%. In a 40×40 complex grid environment, the GACOMR algorithm achieves an optimal path length of 58.84 m, approximately 21.2% shorter than the 74.66 m of the traditional ant colony algorithm, and requires only 7 iterations to converge, whereas the comparison algorithms fail to find a global optimal solution. Additionally, in complex environments, the number of turns in the path generated by the GACOMR algorithm is significantly reduced from 33 to 22, a decrease of 33.3%, making it more suitable for practical AGV navigation requirements. Finally, path planning tests conducted with an AGV in a real-world environment verify the feasibility and effectiveness of the GACOMR algorithm.

Conflicts of Interest

The authors declare that they have no conflict of interest.

References

[1] Yao M, Deng H, Feng X, et al.Improved dynamic windows approach based on energy consumption management and fuzzy logic control for local path planning of mobile robots[J]. Computers & Industrial Engineering, 2024(Jan.):187.

[2] Qu Dan, Qi Lei, Wang Xing. Research on Optimization of Electric Refrigerated Vehicle Distribution Routes Based on Improved Ant Colony Algorithm[J]. Journal of Zhengzhou University of Aeronautics, 2025, 43(02): 104-112.

[3] Liu Li, Liu Yong, Sun Yunquan, et al. AGV Path Planning Optimization Based on Adaptive Ant Colony Algorithm[J]. Electronic Measurement Technology, 2023, 46(18): 100-107.

[4] Jin Jiang, Wang Xiaoping, Zang Tiegang, et al. Robot Obstacle Avoidance Path Planning Based on Improved Ant Colony Algorithm[J]. Computer Engineering and Design, 2025, 46(04): 950-958.

[5] Lin Shuo, Jin Hengjiang, Han Zhonghua, et al. Research on Improved A* Path Planning Algorithm in Random Maps[J]. Manufacturing Technology & Machine Tool, 2025, (02): 43-47.

[6] Zhou Wei, Hu Yi, Liu Jinjiang, et al. Research on Cooperative Scheduling of AMR Clusters Based on Dijkstra Algorithm[J]. Manufacturing Technology & Machine Tool, 2023, (06): 175-179.

[7] Chen Lifang, Yang Huogen, Chen Zhichao, et al. Global Path Planning by Integrating B-Spline Technique and Genetic Algorithm[J]. Journal of Zhejiang University (Engineering Science), 2024, 58(12): 2520-2530.

[8] Sohrabinejad A, Hafshejani K F, Sobhani F M. Leveraging genetic algorithms and system dynamics for effective multi-objective policy optimization: a case study on the broadcasting industry[J]. SIMULATION, 2025.

[9] Li G, Liu C, Wu L, et al. A mixing algorithm of ACO and ABC for solving path planning of mobile robot[J]. Applied Soft Computing, 2023, 148(000):15.

[10] Luo Yi, Shi Yan, Lin Chunsong, et al. Trajectory Optimization of Robotic Arm Based on Mutation Strategy Particle Swarm Optimization Algorithm[J]. Manufacturing Technology & Machine Tool, 2025, (03): 56-64+76.

[11] Ding Linhao, Jiang Hongyan. AGV Path Planning Based on Improved Ant Colony Algorithm[J]. Automation & Instrumentation, 2025, 40(03): 66-70.

[12] Yue Chunlei, Huang Jun, Deng Lele. Research on Improved Ant Colony Algorithm for AGV Path Planning[J]. Computer Engineering and Design, 2022, 43(09): 2533-2541.

[13] Qu Xinhuai, Xu Chenglong, Ding Birong, et al. Research on AGV Path Planning Based on Improved Ant Colony Algorithm[J]. Journal of Hefei University of Technology (Natural Science), 2024, 47(07): 865-869.

[14] Zeng Yuju, Chen Bo, Qu Rui, et al. Research on Mobile Robot Path Planning Based on Improved Ant Colony Algorithm[J]. Modern Manufacturing Engineering, 2023, (10): 57-63+119.

[15] Huo F, Zhu S, Dong H, et al. A new approach to smooth path planning of Ackerman mobile robot based on improved ACO algorithm and B-spline curve[J]. Robotics and Autonomous Systems, 2024, 175(000).

[16] Zhang Zhijun, Dong Xueping, Gan Min. Research on AGV Path Planning Based on Optimized Ant Colony Algorithm[J]. Journal of Hefei University of Technology (Natural Science), 2022, 45(07): 914-919+924.

[17] Wang Xiaoming, Shang Shuo, Liang Chao, et al. AGV Path Planning Using Improved Ant Colony Algorithm[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2023, (03): 63-65.

[18] Zhou Jingdong, Gao Weizhou, Yang Wenguang, et al. Mobile Robot Path Planning Based on Improved Ant Colony Algorithm[J]. Science Technology and Engineering, 2022, 22(28): 12484-12490.

[19] Zhang Tianrui, Wu Baoku, Zhou Fuqiang. Research on Improved Ant Colony Algorithm for Global Path Planning of Robots[J]. Computer Engineering and Applications, 2022, 58(01): 282-291.

[20] Huang Fengyun, Jiang Shiqiu, Xu Jianning. Research on Improved Ant Colony Algorithm for Robot Path Planning[J]. Machinery Design & Manufacture, 2023, (12): 194-198.