

AGV Path Planning based on Improved Ant Colony Algorithm

Haoran Liu

Shandong University of Science and Technology, Qingdao, Shandong, China

Abstract: With the widespread application of Automated Guided Vehicles (AGV) in industrial production and warehouse logistics, the challenges they face during operation are becoming increasingly apparent. Currently, the path planning problem of AGV has become one of the hot topics in academic research. This paper provides an in-depth analysis of the performance of AGV in real-world scenarios and utilizes grid-based methods to construct a map environment model. Subsequently, a detailed analysis of the movement of AGV in the environment is conducted, and an improved pheromone method is proposed based on the traditional ant colony algorithm, aiming to optimize the navigation path of AGV to determine the optimal travel route.

Keywords: AGV; Ant Colony Algorithm; Path Optimization; Grid Map.

1. Introduction

With the advancement of technology, Automated Guided Vehicles (AGV) have been widely utilized in various industries such as intelligent warehousing, smart manufacturing, and logistics transportation due to their high level of automation, flexibility, and other advantages[1]. Currently, AGV have become an indispensable component of intelligent workshop systems.

Addressing the AGV path planning problem, the Ant Colony Optimization (ACO) algorithm has been successfully applied in solving the Traveling Salesman Problem (TSP) and path planning due to its advantages such as parallel processing capability[2], positive feedback mechanism, and good scalability. However, the ACO algorithm suffers from the drawbacks of long computation time and susceptibility to getting trapped in local optima[3]. To address these shortcomings, an improved ant colony algorithm has been proposed, which differentiates the treatment of pheromone concentrations to expedite convergence.

2. Grid-based Method for Modeling

The grid-based method for modeling is a mapping representation technique based on grids. It divides the mapped area into uniform grid cells and assigns specific attribute information to each grid cell, providing an efficient and flexible environment modeling approach for mobile robots [4].

Introducing environmental modeling into a two-dimensional map, AGV is abstracted as a moving point. On the grid map, the path of the Automated Guided Vehicle (AGV) consists of a series of connected blank grid elements, with the starting and ending points of these elements corresponding to the initial and target positions of the task. A Cartesian coordinate system is introduced into the grid map, enabling precise positioning of each grid using coordinate values. The Cartesian coordinate system grid map established using this method uses colors to identify different types of grids: white represents free grids that can be traversed, while black represents obstacle grids that hinder movement.

Below is the construction of the map. First, we establish a 20×20 grid map. The unit length is 1 meter. In this grid map,

the AGV is represented as a point particle capable of moving at a fixed speed. It is treated according to a grid, and during movement, it can only occupy one grid at a time. The constructed grid map model is illustrated in Figure 1.

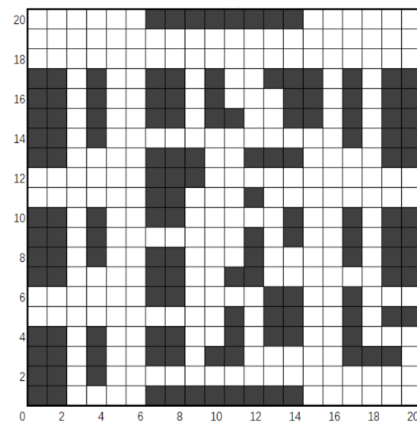


Figure 1. Grid Map Model of AGV.

225° 270° 315°

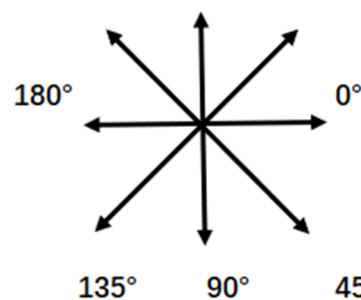


Figure 2. Diagram of AGV Movement Directions

3. AGV Motion Characteristics

In the actual operational environment, AGVs typically move in eight directions, with directions based on 8 cardinal points. These directions are obtained by rotating clockwise from the horizontal X-axis at 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°, as illustrated in Figure 2.

In the grid map, we assume the current position of the AGV is at coordinates $P(x, y)$. Considering the modeling constraints of the grid map, the range of x values is from 1 to

20, and the range of y values is also from 1 to 20. Therefore, the position $P(x, y)$ belongs to the spatial modeling set G . When the AGV moves from time t to the next time $t + 1$, the distance traveled at this moment is denoted as $d(t)$. Assuming the total distance from node i to node j is represented as L , then L is the sum of all $d(t)$ distances. Thus, we derive Equation 1.

$$d(t) = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \quad (1)$$

4. Introduction to Ant Colony Optimization

Ant Colony Optimization (ACO) is a swarm intelligence optimization algorithm that simulates the foraging behavior of ants[5]. The core idea of the ACO algorithm is to mimic the process of ants searching for food in nature. Ants can find the shortest path from their nest to a food source through simple rules and the exchange of pheromones. This algorithm demonstrates excellent performance in solving combinatorial optimization problems such as the Traveling Salesman Problem (TSP) [6], scheduling problems, path planning problems, and others.

The basic principles include:

Pheromone Trails: Ants release a substance called pheromone while searching for paths. Other ants can perceive these pheromones and tend to move along paths with higher pheromone concentrations.

Probabilistic Selection: When facing multiple path choices, ants probabilistically select paths based on the concentrations of pheromones along each path. Paths with higher pheromone concentrations have a higher probability of being chosen.

Pheromone Update: Over time, pheromones on paths that have been traversed gradually evaporate, while ants leave new pheromones on these paths. As a result, the concentration of pheromones on shorter paths tends to be relatively higher.

Collaboration: Through the collaboration of a large number of ants, one or several optimal paths are eventually formed, which represent the solution to the problem. In traditional ant colony algorithms, ants follow the guidance of pheromone concentrations when choosing paths, and pheromone update and evaporation are also integral parts of the algorithm. Below are the formulas for the ant colony algorithm:

Here is the formula for ant k transitioning from node i to j in the traditional ant colony algorithm:

$$p_{kij}(t) = \frac{\tau_{ij}(t)^\alpha \times \eta_{ij}^\beta}{\sum_{l \in allowed_k} \tau_{il}(t)^\alpha \times \eta_{il}^\beta} \quad (2)$$

In the ant colony algorithm, we define several parameters. Firstly, we define $\tau_{ij}(t)$ to represent the residual pheromone concentration on the path (i, j) at time t , initially set as $\tau_{ij}(t) = C$ (where C is a constant). Then, $p_{kij}(t)$ denotes the probability of ant k transitioning from node i to node j . In this probability calculation, we typically use η_{ij} as the heuristic factor. Additionally, we introduce α to denote the relative importance of residual pheromones on path (i, j) , and β represents the importance of the heuristic factor. Furthermore, $allowed_k$ denotes the set of paths that ant k is allowed to choose from at time t .

After one iteration, we calculate the length L_k of the paths traveled by each ant and compare to determine the shortest path L_{min} .

Assuming the ants have completed one iteration, the update rule for the pheromone on each path is as follows:

$$\tau_{ij}(t + 1) = (1 - \rho) \cdot \tau_{ij}(t) \quad (3)$$

Where $\tau_{ij}(t)$ represents the residual pheromone concentration on path (i, j) .

The amount of pheromone left by each ant on the path is related to the performance of the path that the ant has traversed. Typically, ants that perform well on a path leave behind more pheromones. This update can be performed after each ant completes one iteration.

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

Q is a constant representing the amount of pheromones left by ants on the path. It is the total length of the paths traversed by the k -th ant in the current iteration. Combining evaporation and updating, we can obtain the overall update formula for the pheromone concentration on each path:

$$\tau_{ij}(t + 1) = (1 - \rho) \times \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (5)$$

Thus, after all ants complete one iteration, the pheromone concentration on each path can be updated according to the above formula.

5. Differential Update of Pheromones

In the traditional ant colony algorithm, ants optimize their route selection based on the level of pheromones. However, this method may lead to slow convergence of solutions. Therefore, to improve the convergence speed of the algorithm and save search time, we need to confine the search scope to advantageous paths. After each iteration, we update the difference in pheromone levels between the optimal path and the worst path. This allows ants to focus more on advantageous paths, increasing the likelihood of finding the optimal solution near these paths.

Updating the pheromone level on the optimal path:

$$\Delta\tau_{best} = Q \quad (6)$$

Updating the pheromone level on the worst path:

$$\Delta\tau_{worst} = -Q \quad (7)$$

This way, after each iteration, the optimal path will receive positive reinforcement of pheromones, while the worst path will receive negative inhibition of pheromones. Then, we apply these updates to the traditional pheromone update formula:

$$\tau_{ij}(t + 1) = (1 - \rho) \times \tau_{ij}(t) + \Delta\tau_{ij} \quad (8)$$

Where $\Delta\tau_{ij}$ is the change in pheromone level for path (i, j) , which depends on the type of path (either the best path or the worst path). For example: if path (i, j) is the best path, then $\Delta\tau_{ij} = \Delta\tau_{best}$. If path (i, j) is the worst path, then $\Delta\tau_{ij} = \Delta\tau_{worst}$. For other paths, $\Delta\tau_{ij}$ can be calculated based on factors such as the length of the path traversed by the ants. This way, ants will be more inclined to search near the best path, accelerating the convergence speed of the algorithm and saving search time.

In the initial stages, the best path is typically distant, resulting in relatively low gains in pheromone, which favors maintaining diversity in path selection. However, as iterations progress and the best path gradually approaches the goal, the gain in pheromone increases, thereby enhancing search efficiency. Similarly, for the worst path, there is relatively less

decay of pheromones in the initial stages to prevent premature neglect of path information. As iterations progress and the worst path approaches the goal, the decay of pheromones increases. Therefore, for each segment of the best and worst paths, the changes in pheromone are different, thus amplifying the differences between the best and worst paths.

6. Global Pheromone Adjustment

In the ant colony algorithm, after each iteration, the pheromone update is performed only based on the best and worst paths, which to some extent accelerates the convergence speed. However, in the initial stages of the algorithm, ants often prematurely focus on several suboptimal paths, potentially overlooking the possible optimal path, thus hindering the discovery of the optimal solution. To prevent ants from prematurely concentrating on a few suboptimal paths in the early stages of the algorithm, measures can be taken for paths other than the best and worst paths.

Introducing an exploration factor γ allows ants to not fully rely on the current pheromone concentration when selecting paths. Therefore, the pheromone concentration τ_{ij} in the probability calculation formula for paths is multiplied by the exploration factor γ . This allows for a certain probability of randomly selecting non-optimal paths, helping ants explore the search space more extensively.

$$p_{kij}(t) = \frac{(\tau_{ij}(t)^\alpha \times \eta_{ij}^\beta)^\gamma}{\sum_{l \in allowed} (\tau_{il}(t)^\alpha \times \eta_{il}^\beta)^\gamma} \quad (9)$$

Where γ is the exploration factor, controlling the degree of exploration. A larger value of γ increases the likelihood of exploration, which benefits the ant colony in discovering the optimal path.

7. Flowchart of the Improved Ant Colony Algorithm

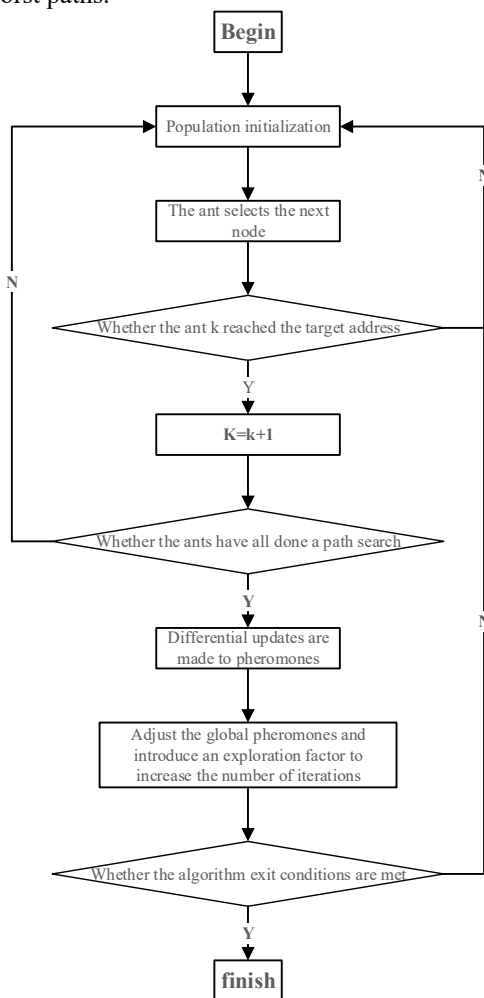


Figure 3. Flowchart of the Improved Ant Colony Algorithm
Steps of the Improved Ant Colony Algorithm

- Initialize the population.
- Move ants to the next node based on the ant colony algorithm.
- Check if the ant reaches the destination point, and record the path length of ant k .
- Update the concentration of pheromones on the paths by altering the differences in pheromone levels.
- Improve the exploration factor in the early stages of the algorithm by updating the pheromones on other paths to ensure that ants do not prematurely focus on a few suboptimal

paths.

If the optimal path for each ant is consistent, then the algorithm is complete.

Output the shortest path and the number of iterations, and exit the program.

8. Simulation Experiment Comparison

By utilizing the MATLAB program of version 2022b, a comparison output was conducted between the basic ant colony algorithm and the improved ant colony algorithm,

with the same set of parameters: $\alpha = 1$, $\beta = 6$, $m = 100$, $\rho = 0.4$, $k = 200$, with the starting coordinates at (1,20) and the target coordinates at (20,1). By running these two different algorithms, we obtained their respective optimal path lengths as shown in Figure 4 and Figure 5, and the number of iterations as depicted in Figure 6 and Figure 7.

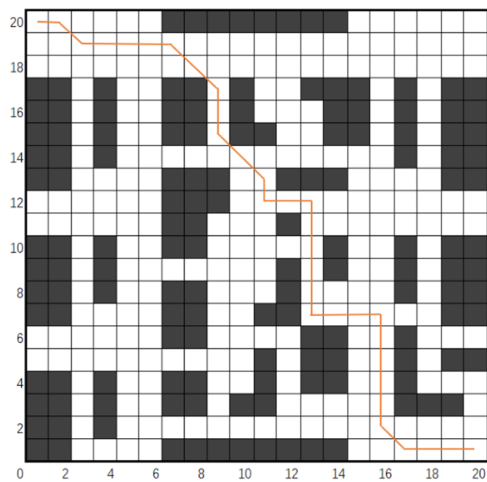


Figure 4. Diagram of Paths Before Improvement

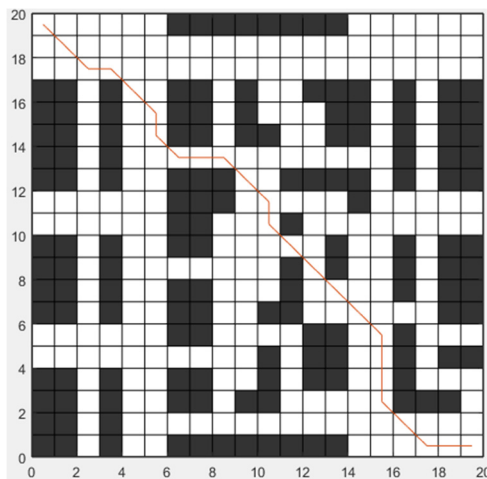


Figure 5. Diagram of Paths After Improvement.

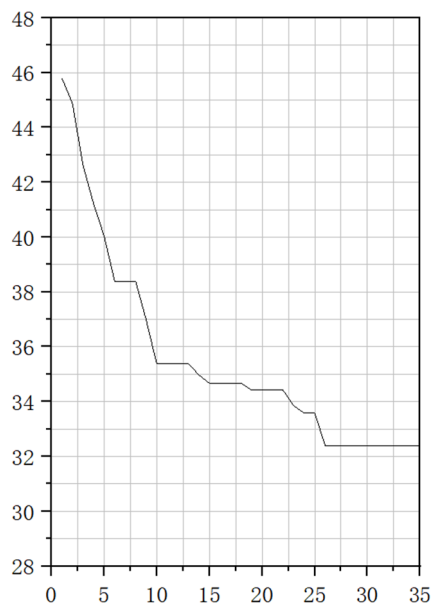


Figure 6. Improved pre-iteration curves

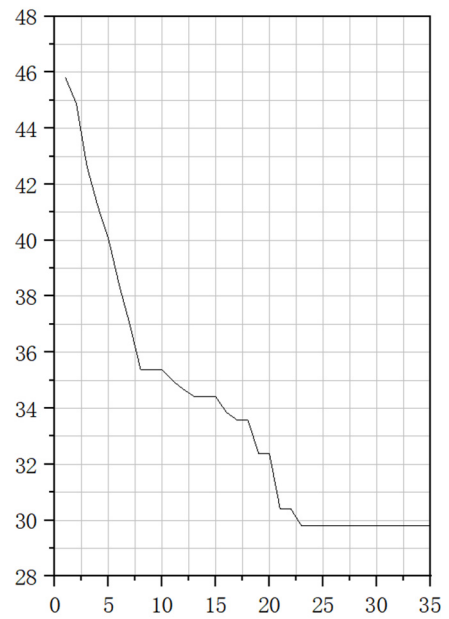


Figure 7. Improved iterative curves.

Through the comparison of the images output by MATLAB, we can see that the improved ants are significantly better than the traditional ant colony algorithm in the optimal path selected, and in the iterative image, the convergence speed of the significantly improved ant colony algorithm is faster than that of the traditional ant colony algorithm, and all ants reach the optimal solution.

Therefore, the traditional ant colony algorithm is compared with the improved ant colony algorithm, and the results are shown in Table 1:

Table 1. Comparison before and after algorithm improvement

performance	Traditional ant colony algorithms	Improved ant colony algorithm
Optimal path	32.38m	29.80m
The number of iterations	28	25

By comparing the results, we can see that the search efficiency of the traditional ant colony algorithm is low, and it is prone to suboptimal paths. The improved ant colony algorithm has higher search efficiency, shorter time, and improved the accuracy of finding the optimum.

9. Conclusion

This paper improves the ant colony algorithm by enhancing the differentiation in pheromone distribution, thereby increasing the speed of ants in searching paths and accelerating convergence. Additionally, it enhances the exploration factor of global pheromone adjustment, which improves the accuracy of the search map. Through simulation experiments, the results demonstrate the feasibility and effectiveness of the method.

References

- [1] Liu Jiaqi, Wang Taihua, Dong Zheng. Mobile Robot Path Planning Based on Improved Ant Colony Algorithm[J]. Sensors and Microsystems, 2022, 41(5): 140-143.
- [2] Guo Shikai, Sun Xin. Mobile Robot Path Planning Based on Improved Particle Swarm Algorithm[J]. Electronic Measurement Technology, 2019, 42(3): 54-58.

- [3] Chen Hu, Song Zhichao, Guang Mengke, et al. Path Planning of Mobile Robots in Complex Environments[J]. Journal of Guangxi University (Natural Science Edition), 2021, 46(3): 692-702.
- [4] Gu Jun-hua, Fan Pei-pei, Song Qing-zeng. Improved culture ant colony optimization method for solving TSP problem. computer engineering and Applications[J]. Computer Engineering and Applications, 2010(26):49-52.
- [5] ROKBANI N, KUMAR R, ABRAHAM A, et al. Bi-heuristic ant colony optimization-based approaches for traveling salesman problem [J]. Soft Comput, 2021, 25(5):3775-3794.
- [6] Li Jun, Liu Guang-ru. Path planning of mobile robot based on improved ant colony algorithm[J]. Machinery Design & Manufacture, 2010(8):164-165).