Computer Fraud and Security
ISSN (online): 1873-7056

Improved Ant Colony Algorithm in Optimizing Evacuation Path Planning

Yu Guo, Yuhang Sun, Bin Chen, Xiaobin Jiang*
Merchant Marine College, Shanghai Maritime University, Shanghai 201306, China
**Corresponding Author.*

Abstract:
The traditional ant colony algorithm has played an important role in the evacuation path planning of ship personnel. By simulating the process of ants searching for food, it effectively solves the optimization problem of evacuation paths, improves evacuation efficiency and safety. However, traditional ant colony algorithm in path planning applications also has some shortcomings, such as the heuristic function does not introduce the distance of the target point, insufficient utilization of obstacle information, and insufficient difference in the probability of each node. Therefore, this paper proposes an improved ant colony algorithm to solve these problems. In view of that the traditional ant colony algorithm converges slowly and easily falls into local optimum, the distance correction function is introduced into the heuristic function by combining the basic concept of the evaluation function of A* algorithm; the obstacle avoidance function is introduced to make the heuristic function guide the ants forward more reasonably; the smoothness function is introduced to reduce the number of turns and control the turning angle of the planned path. The effectiveness and applicability of the improved algorithm are verified through two sets of experiments, and the improved ant colony algorithm meets the needs of emergency evacuation of actual ship personnel. Simulation and experimental results show that the method proposed can solve the randomness of turning path selection at the early stage of the algorithm, has better global search ability and convergence, and performs better in the number of iterations and computation time, and can provide efficient and safe evacuation plan in the shortest time, which meets the demand of emergency evacuation of actual cruise ship personnel.

Keywords: improved ant colony algorithm, ship evacuation path, heuristic function, smoothness function.

INTRODUCTION

Since the end of the last century, the world cruise tourism economy has been growing at an average annual rate of 8%, reaching two times the level of economic growth of the global tourism industry, known as "floating on the golden waterways of the gold industry". Compared with other ships, the internal structure of the cruise ship is more complex, at the same time, the number of passengers on board, and most of the passengers have not gone through professional survival training, how to better cope with the evacuation of personnel in an emergency has become a hot issue. In order to minimize casualties, accurate, fast and effective evacuation path planning is particularly important in the event of an accident. Path planning algorithm has been the core and hot issue of evacuation problem, with the research objective of finding an optimal path or better path between two points. The advantages and disadvantages of the path are mainly in terms of travel length, smoothness and safety. After decades of development, scholars at home and abroad have proposed many fruitful path planning algorithms, such as A* algorithm [1], genetic algorithm [2], particle swarm algorithm [3], artificial potential field method [4], neural network [5], Q-Learing algorithm [6], etc. These methods have achieved better results in path planning, but still have certain defects when facing complex environments.

The traditional ant colony system (ACS) is a probabilistic algorithm used to find optimized paths. It was inspired by Marco Dorigo's observation of ants' behavior in finding paths while searching for food, and was proposed in his doctoral thesis in 1991. This algorithm has advantages such as distributed computing, positive information feedback, and heuristic search [7], but it also has disadvantages in path planning applications such as low search efficiency and susceptibility to getting stuck in local optimum. To address these problems, many scholars have improved ACS for path planning and achieved better results: literature [8] optimized the pheromone recording rules and related parameter values of the traditional ACS algorithm to reduce the probability of iterative stagnation and premature convergence of the algorithm and improve the solving ability of the algorithm; literature [9] proposed a new pheromone update algorithm and global/local pheromone distribution model to achieve fast search for the shortest path, thus improving the efficiency of path planning; literature [10] proposed a historical experience-guided pheromone update method, which adaptively performs different types of pheromone update schemes according to the search history to improve the algorithm search performance; literature [11] proposed an improved heuristic function and pheromone update mechanism considering path length and slope smoothing factors to improve the global search capability and convergence of the algorithm; In literature [12], the search efficiency of the algorithm was improved by limiting the horizontal and vertical search directions of ants, and the adaptive expectation function and heuristic factor were introduced to dynamically adjust the state transfer probability to improve the convergence speed of the algorithm. The literature [13] used the raster map method to change the transfer probability of the ACO and added the dead zone judgment to reduce the number of "invalid ants" to accelerate the convergence speed and improve the stability of the algorithm.

To summarize the existing research, it can be seen that most of the above improvement methods of ant colony algorithm focus on improving the search efficiency of ant colony algorithm, reducing the number of iterations and getting the shortest possible path, but there is less research on other factors (such as the number of turns and turning angles) that affect the merit of the path. Based on this, this paper introduces a distance correction function in the heuristic function for the disadvantages of slow convergence and easy to fall into local optimum of the traditional ant colony algorithm, combined with the basic concept of the evaluation function of A* algorithm [14]; an obstacle avoidance function is introduced to make the heuristic function better guide ants forward; a smoothness function is introduced to reduce the number of turns and control the turning angle of the planned path.

OBJECTIVES

To address some shortcomings of the traditional ant colony algorithm in path planning applications, such as the heuristic function does not introduce the distance of the target point, insufficient utilization of obstacle information, and insufficient difference in the probability of each node, this paper proposes distance correction, obstacle avoidance function and smoothing function for improvement through a multi-factor integrated heuristic function, while improving the pheromone update model to make the planned path smoother, and finally getting the improved ant colony algorithm.

METHODS

Ant Colony Algorithm

Ant colony algorithm can be regarded as the process of ant populations starting from a common point and going to an unknown location to find food. Each ant's decision on its direction mainly depends on two key factors: pheromone concentration and heuristic information. Pheromones are equivalent to directional information emitted by a group. The higher the concentration of pheromones on a path, the more it can guide ants to walk along that path. Heuristic information is the self judgment information of each ant based on its own environment. Ants determine how to transfer next by combining individual and group information. Therefore, the construction of pheromone concentration models and heuristic functions is the key to the superiority or inferiority of ant colony algorithms.

How the ant chooses its next direction of travel is based on the transfer probability derived from the following equation.

$$p_{i,j}^{\mathrm{k}}(t)=\begin{cases}\dfrac{\left[\tau_{i,j}(t)\right]^{\alpha}\left[\eta_{i,j}^{k}(t)\right]^{\beta}}{\sum\limits_{s\subset \text{allowed}_i}\left[\tau_{i,j}(t)\right]^{\alpha}\left[\eta_{i,j}^{k}(t)\right]^{\beta}}, j\in \text{allowed}_i \\ 0, j\notin \text{allowed}_i\end{cases} \tag{1}$$

where $p_{i,j}^{\mathrm{k}}(t)$ denotes the probability of the k th ant transferring from the current grid i to the neighboring grid j at the t th iteration, τ denotes the pheromone, η denotes the heuristic function, α , β denote the relative importance of the pheromone and the heuristic factor, respectively, and allowed_i denotes the set of the next transferable neighboring grid of the current grid i , where $\eta_{i,j}^{k}(t)$ is inversely proportional to the Euclidean distance between two grids i , j . The larger the Euclidean distance D_{ij} between two grids i , j , the smaller the probability that ant k picks that path, which is usually.

$$\eta_{i,j}^{k}(t)=\frac{1}{D_{ij}} \tag{1}$$

where D_{ij} denotes the Euclidean distance between grids i, j.

Pheromone is the chemical information left by ants in the process of traveling to guide the subsequent ants. The shorter the path the ants travel, the more dense the pheromone will be, and the more it can guide the ants to take a shorter path. With the growth of the number of iterations, the pheromone will accumulate and volatilize, therefore, a pheromone update model is needed, with the following equation.

$$\begin{cases}\tau_{\mathrm{ij}}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij},0<\rho<1\\ \Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}\end{cases} \tag{2}$$

$$\Delta\tau_{i,j}^{k}(t)=\begin{cases}Q/L_{\mathrm{k}},\{i,j\}\subset \text{ visited }_{t,\mathrm{q}}^{k}\\ 0,\ \text{others}\end{cases} \tag{3}$$

where $\Delta\tau_{ij}^{k}$ is the pheromone concentration released by the k th ant on the path between grids i, j, $\Delta\tau_{ij}$ is the sum of the pheromone concentrations released by all ants on the path between grids i, j. ρ is the pheromone volatility coefficient, which generally takes the value $0<\rho<1$, Q is the pheromone constant, L_{k} is the total length of the path traveled by ant k in the t th iteration, m is the total number of ants, $\text{visited}_{t,\mathrm{q}}^{k}$ is the ordered set of grid labels visited by ant k up to grid q in the t th iteration, and q is the target grid number.

The initial pheromone is usually set as follows.

$$\tau_{ij}(0)=C \tag{4}$$

C is the initial pheromone constant.

Multi-factor Integrated Heuristic Function

At the early stage of the algorithm operation, because the initial pheromone is a fixed value, the difference of pheromone values among the grids is small, the inspiration is weak, and the choice of ant travel direction is random, which not only affects the convergence speed of the algorithm, but also easily goes into a dead end, and the search efficiency of the algorithm is not high. In order to make the heuristic function a better guide for the ants, the following changes are made to the heuristic function.

$$\eta_{i,j}^{k}(t)=D(i,j,q)+\partial_{i,j}^{k}(t)+N^{k}(t) \tag{6}$$

where $D(i,j,q)$ is the distance correction function, $\partial_{i,j}^{k}(t)$ is the direction correction function, and $N(i,j)$ is the smoothing function.

Distance correction function

The heuristic function of traditional ant colony algorithm is only related to the distance between adjacent grids. The difference in values is small, the inspiration is weak, and the search efficiency of the algorithm is not high. Combined with the basic concept of the evaluation function of A* algorithm[14], this paper constructs the distance correction function of the improved ant colony algorithm, and introduces the predicted distance between the current grid and the target grid on the basis of the original heuristic function. The distance correction function is introduced as follows.

$$D(i,j,q)=\frac{1}{\lambda_1 D_{ij}+\lambda_2 D_{jq}}*\frac{D_{\max}-D_{jq}}{D_{\max}-D_{\min}} \tag{7}$$

In equations (6)-(7), $D(i, j, q)$ is the distance correction function, which indicates the corrected distance from a neighboring raster j of the current raster i to the target raster q, where λ_1 and λ_2 ($0 \le \lambda_1 \le 1$, $0 \le \lambda_2 \le 1$) are correction parameters, which can be decided according to the specific application environment, $D_{\max}$ and $D_{\min}$ denote the maximum and minimum values of the Euclidean distances from all neighboring rasters of raster i to the target raster, D_{ij} indicates the Euclidean distance between raster i and neighboring raster j, and D_{jq} indicates the Euclidean distance between neighboring raster j and target raster q.

Obstacle avoidance function

At the early stage of algorithm operation, the ants' travel direction may be very chaotic, and the possibility of going into a dead end is great, and the situation that ants cannot reach the end point, i.e., the deadlock phenomenon, is not conducive to guiding the subsequent ants to find the optimal path. The reason for this phenomenon is that the ants are limited by the U-shaped obstacles in the raster map and their own trajectories. However, due to the limited number of individuals in the ant population, giving up some ants will not only have an impact on the algorithm's ability to find the best, but also reduce the convergence speed of the algorithm. In order to make the heuristic function better guide the ants forward, the obstacle avoidance function is introduced as follows.

$$\partial_{i,j}^{k}(t) = \frac{allowed_{i,j}^{k}(t)}{all_{i,j}^{k}(t)} \tag{8}$$

In equation (8), $allowed_{i,j}^{k}(t)$ is the number of transferable grids in the neighboring grids of grid *i*, $all_{i,j}^{k}(t)$ is the total number of neighboring grids of grid *i*, and $\partial_{i,j}^{k}(t)$ is the obstacle avoidance function, which is used to calculate the proportion of transferable grids in the neighboring grids, indicating that the more obstacles around the neighboring grids, the smaller the probability of ants *k* choosing the path, and vice versa, so as to quickly guide the ants to choose the optimal direction of travel and obstacle avoidance, and also to reduce the probability of ants walking into dead ends and improve the algorithm convergence speed.

Smoothing functions

The goal of path planning in ship personnel evacuation is to minimize the evacuation time of personnel. However, the personnel evacuation time is not necessarily proportional to the navigation distance when considering the effect of the number of turns and the angle of the turns on the walking speed of the crowd. For constant condition sections, the energy consumed to complete the task is directly proportional to its navigation time. Therefore, the problem of two-dimensional smoothness is considered in path planning, and the number of turns and turning angles of the path is desired to be as small as possible, and the two-dimensional smoothness heuristic function is introduced with the following equation

$$N^{k}(t) = \lambda_3 \sum_{n=1}^{q} \varphi(t)_n^k + \lambda_4 \sum_{n=1}^{q} \theta(t)_n^k \tag{9}$$

In equation (9), $N^{k}(t)$ is the smoothing function, which indicates the number of turns of ant k in the tth iteration, $\sum_{n=1}^{q} \varphi(t)_n^k$ indicates the total number of turns of ant k visiting the grid so far, $\sum_{n=1}^{q} \theta(t)_n^k$ indicates the total angle of turns of ant k visiting the grid so far, λ_3, λ_4 ($0 \le \lambda_3 \le 1$, $0 \le \lambda_4 \le 1$) are the correction parameters, which can be decided according to the specific application environment, if the turn of this iteration is the same as the last turn, the heuristic function is larger according to the turn, and vice versa is smaller, which can guide ants to follow a straight line as much as possible during their movement, avoiding the zigzag path when the grid distribution is too dense.

Improving the Pheromone Update Model

In practice the optimal path not only requires a short path, but also more smooth, therefore, this paper uses the following methods to improve the pheromone update model.

$$\eta_{ij}(t)=\frac{1}{D_{ij}+d_{jE}} \tag{10}$$

$$Q=C \tag{11}$$

$$S_m(t)=\lambda_5 L_m(t)+\lambda_6 N_m(t) \tag{12}$$

In equation (12), $S_m(t)$ is the integrated index of the *m*th ant walking through the path in the *t*th iteration, $L_m(t)$ is the path length, $N_m(t)$ is the number of turns. The integrated index determine the allocation method of pheromones, the smaller the index, the better the path. λ_5 , λ_6 are the adjustment coefficients of each factor, which are appropriately valued according to the nature of the desired path.

Improved ant Colony Algorithm Flow

Figure 1 shows the flow of the improved algorithm proposed in this paper.

Step 1: Establish an environment model based on planar graphics and initialize algorithm parameters.

Step 2: Determine the pathfinding starting point and ending point grid, and place m ants at the starting point grid.

Step 3: Calculate the distance correction function, obstacle avoidance function, and smoothing function for each raster by equations (7), (8), and (9) based on the map and obstacles, and then calculate the multi-factor integrated heuristic function based on equation (6).

Step 4: Calculate the transfer probability based on the pheromone and the comprehensive heuristic function of multiple factors, select the grid that the ant will walk next, and add the grid that has been walked to the taboo table.

Step 5: Determine whether the ant has reached the end grid, if so, construct the complete path of the ant, otherwise, return to Step 4.

Step 6: Calculate the length $L_m(t)$ and the number of turns $L_m(t)$ of the path taken by the ant, and find the optimal path for this iteration.

Step 8: Compare the optimal paths of each iteration and find the current optimal path.

Step 9: Determine whether the maximum number of iterations T_{max} is reached, if yes, output the current optimal path and end the algorithm; otherwise, continue to execute the algorithm.

Step 10: Calculate the new pheromones on each path according to equations (3) and (12), and then update the pheromones of the whole map, reset the ants and start the next round of solving.

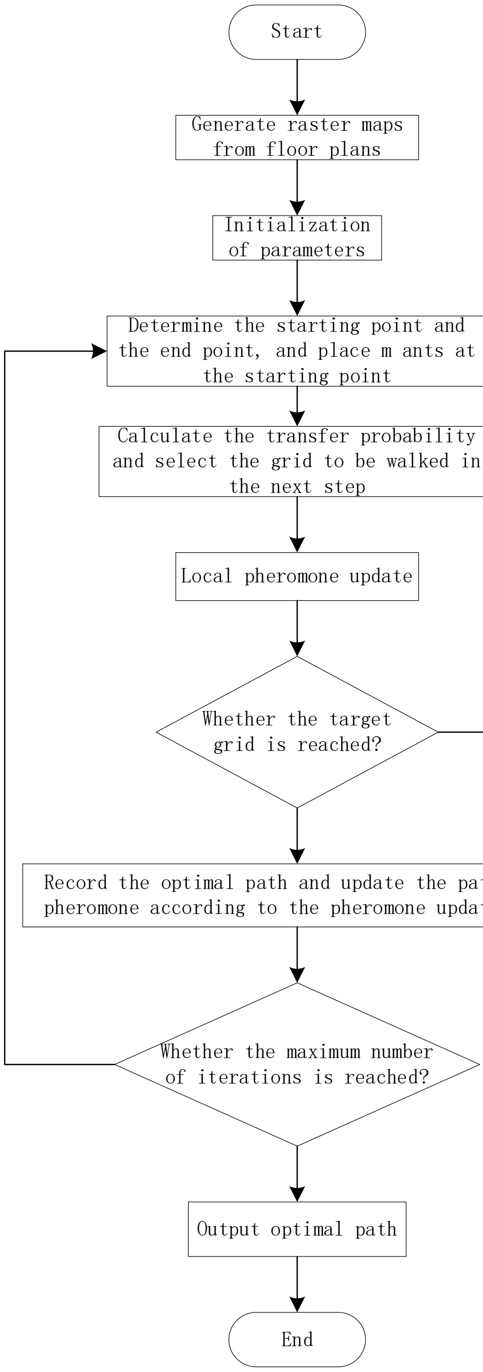


Figure 1. Improved ant colony algorithm flow

RESULTS

To verify the reliability and the advantages and disadvantages of the algorithm proposed, the path planning process of ship personnel in emergency situations were simulated with the traditional ant colony algorithm, the algorithm of literature [11] and the improved ant colony algorithm seperately by Matlab. Since the ant colony algorithm uses the roulette wheel method in selecting the next pending grid, the results calculated in each simulation will be slightly different. Therefore, the algorithms in each experiment are simulated several times in the simulation, and the centered results are taken.

Environment Modeling

A deck plan of the daily living area of a large ship as shown in Figure 2 is selected. The cabin equipment and space structure are simplified and a 1:1 two-dimensional grid diagram model is built in Matlab, as shown in Figure 3.

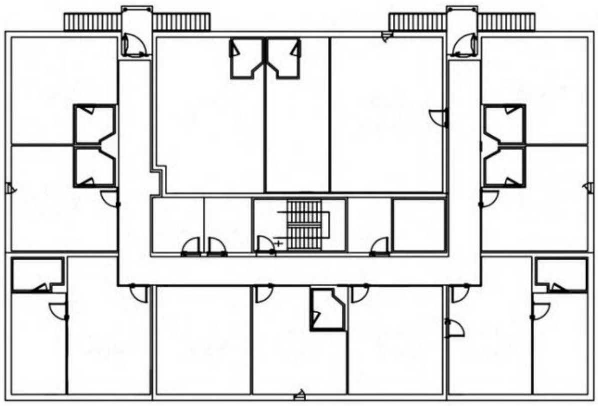


Figure 2. Plan view of the simplified deck

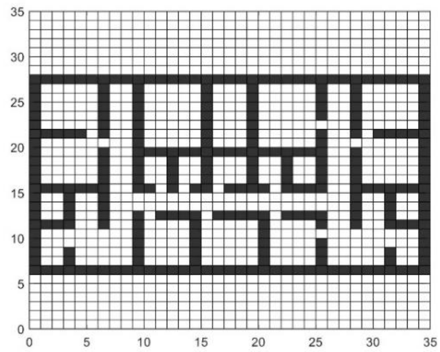


Figure 3. Grid diagram model of the deck

Parameter Selection

The ant colony algorithm involves several parameters that have an important influence on its comprehensive performance, but currently it is not possible to determine the optimal combination of parameters based on theoretical methods, and the optimal parameters are generally obtained through experiments. In the simulation experiments, the default values of other parameters are set as follows: ρ =0.4, m =50, Q =100, T_{max} =100, C =10, λ_1 =0.1, λ_2 =0.01, λ_3 =1, λ_4 =0.01, λ_5 =0.1, λ_6 =1. In order to analyze the impact of different parameters on the performance of the improved ant colony algorithm, simulation experiments under different parameter combinations were conducted on the basis of the change of above taken values for the main parameters α and β of the algorithm. To get representative results, conducted 10 experiments for each parameter combination and took the average value of the solution results of the algorithm. The specific experimental results are shown in Table 1.

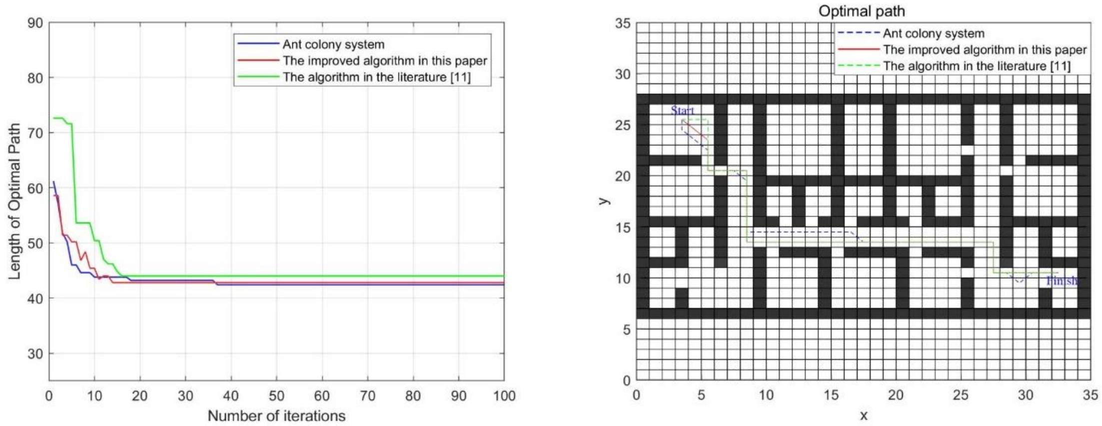
Table 1. The influence of parameters α and β on the algorithm

Parameters	Value	Path length	Number of turns
α	0.1	45.8	16.6
	0.5	43.84	8.8
	1	43.34	7.3
	1.5	43.6	9.5
	2	43.74	9.9
β	1	64.4	39.5
	2	43.34	7.3
	3	43.16	7.2
	4	43.22	7.6
	5	43.4	7.3

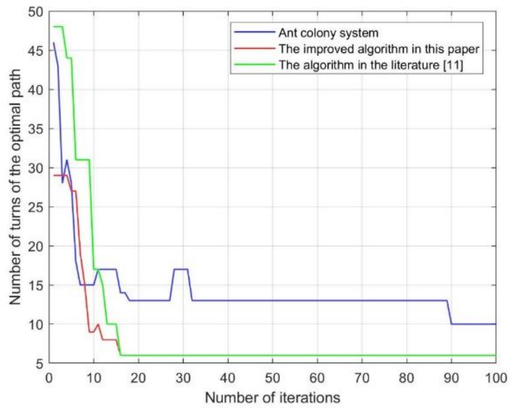
From Table 1, it can be seen that parameters α and β have a significant impact on the algorithm results. Pheromone factor α represents the influence of information quantity on whether to choose the current path, reflecting the relative importance of the information accumulated by ants during movement in guiding the ant colony search. The size of α reflects the strength of the randomness factor in the ant colony's path search. The larger its value, the greater the likelihood that ants will choose previously traveled paths, and the randomness of the search will decrease. When the value of α is too small, it is easy to cause the ant colony's search to prematurely fall into local optimum. By comparing the calculation results, when α was taken around 1, better optimal path length and optimal turning times can be obtained. The heuristic function factor β indicates the direction of pheromone on the path when searching to guide ants to choose the path. The magnitude of β reflects the apriority and strength of the deterministic factors in the process of ant colony searching for the optimal path. The larger the beta parameter is, the more likely the ant is to choose the local shortest path at the local point, which helps speed up the convergence of the algorithm, but also reduces the randomness of the ant colony search for the optimal path, and increases the risk of the search falling into the local optimal solution. On the contrary, if the value of β is too small, the ant colony will search randomly and prefer to determine the path according to the pheromone concentration, so the algorithm converges faster and it is difficult to find the optimal solution. By comparing the calculation results, when β is taken around 2, better optimal path length and optimal turning times can be obtained. Therefore, α=1 and β=2 were taken.

Algorithm Simulation Comparison

The three algorithm are compared and experimented on a map of 35x35 scale. The starting coordinates was set (3.6, 25.2) and the ending coordinates was set (32.4, 10.3). The simulation experiments results are shown in Figure 4 and Table 2.



(a) Optimal path diagram (b) Iterative graph of the length of the path



(c) Iterative diagram of the number of turns of the path

Figure 4. (3.6, 25.2)→(32.4,10.3) Simulation results

Table 2. Simulation results of (3.6, 25.2)→(32.4,10.3)

Algorithm	Ant colony algorithm	The improved ant colony algorithm in this paper	The algorithm in the literature [11]
Optimal path length	42.4	42.8	44
Optimal number of turns	13	6	6
Optimal turning angle	810	495	540
Number of convergence iterations	34.5	15	16
Algorithm running time	3.49	3.99	4.19

In Figure 4, blue, red and green represent the simulation results of the traditional ant colony algorithm, the improved algorithm of this paper and the literature algorithm, respectively; the number of convergence iterations is the average of the convergence iterations for the optimal path length and the convergence iterations for the optimal turning path. According to the simulation results in Figure 4, combined with Table 2, the following conclusions can be drawn.

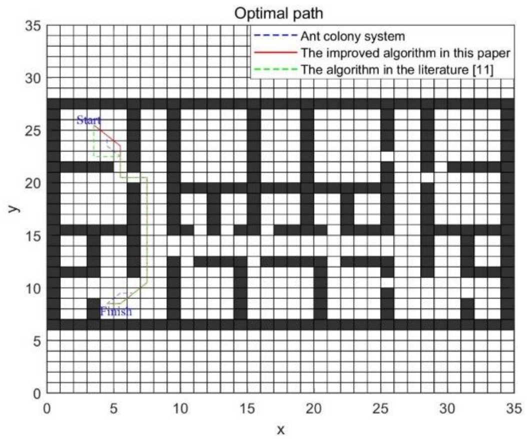
(1) In terms of path length, the improved algorithm of this paper has little difference with the other two algorithms. The traditional ant colony algorithm performs the best, followed by the improved algorithm in this paper, both of which are better than the literature [11] algorithm.

(2) In terms of iteration stability, it is obvious that the advantage of improved algorithm of this paper is very obvious. The previous iteration of the traditional ant colony algorithm and the literature algorithm is very unstable, while the algorithm in this paper benefits from the introduction of obstacle avoidance function in the heuristic function, the overall iterative curve shows a downward trend. The number of stable iterations in the later stage is significantly less than the other two algorithms.

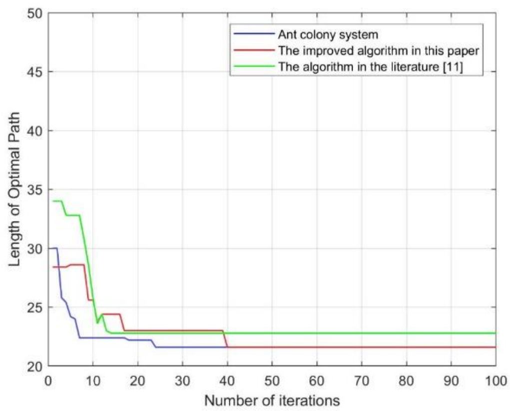
(3) In terms of the number of turns and turning angle of the path, both the improved algorithm in this paper and the algorithm in literature [11] perform well, but the former outperforms the other two algorithms in terms of iterative fluctuation stability. The convergence curve of traditional ant colony algorithm is apparently unstable.

Therefore, in the pursuit of better performance in all aspects of the path, the algorithm in this paper has obvious advantages.

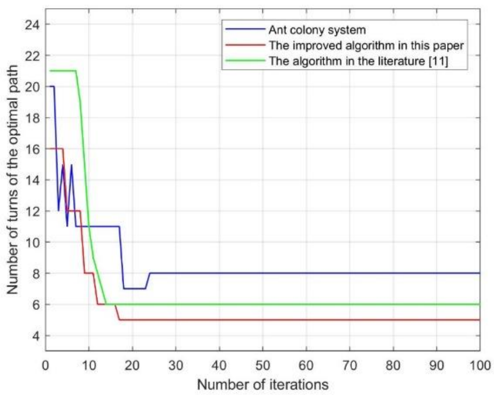
To further verify the reliability of the algorithm, the map of 35x35 scale was used again. Set the starting oordinates (3.6, 25.2) and the ending coordinates (4.4, 8.3). The simulation experiments results are shown in Figure 5 and Table 3.



(a) Optimal path diagram



(b) Iterative graph of the length of the path



(c) Iterative diagram of the number of turns of the path

Figure 5. (3.6, 25.2)→ (4.4, 8.3) Simulation results

Table 3. Simulation results of (3.6, 25.2) → (4.4, 8.3)

Algorithm	Ant colony algorithm	The improved ant colony algorithm in this paper	The algorithm in the literature [11]
Optimal path length	21.6	21.6	22.8
Optimal number of turns	8	5	6
Optimal turning angle	450	315	450
Number of convergence iterations	24	23.5	14
Algorithm running time	2.68	2.78	3.28

According to the simulation results in Figure 5, combined with Table 3, the following conclusions can be drawn.

(1) As shown in Figure 5(a), most of the optimal paths obtained by the 3 algorithms have the same sections. But through comparison, the path of the algorithm in literature [11] has a small number of redundant sections out of the starting point, while the path of the traditional ant colony algorithm has multiple redundant sections, mainly concentrated in the starting point and the path turns, which indicates that the other two algorithms fall into local optimum.

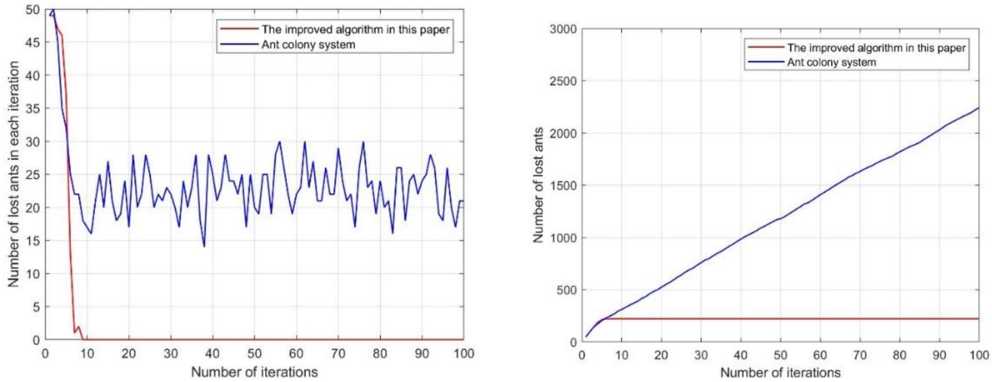
(2) The convergence curves shown in Figure 5(b) and (c) shows that the traditional ant colony algorithm searches for the optimal path with the largest number of turns and converges slowly, obtaining the optimal path with the length of 21.6m and the number of turns of 8 at the 24th iteration. While the literature [11] algorithm converges quickly, however, the length of the optimal path searched is the longest, and the optimal path with a length of 22.8m was obtained in the 14th iteration. Compared with the first two algorithms, the improved algorithm in this paper searches for the shortest optimal path length and the least number of turns, and the convergence speed of the algorithm is in the middle.

(3) In terms of algorithm running time, there is little difference between the traditional ant colony algorithm and the improved algorithm in this paper. The traditional ant colony algorithm performs best, followed by the improved algorithm in this paper, both of which are better than the literature [11] algorithm.

Lost Ants Number Analysis

Lost ants number is an important index to evaluate the comprehensive performance of ant colony algorithm, the smaller, the more accurate the optimal solution will be and the stronger the positive feedback of information; while the larger, it may lead to some never-searched paths with zero pheromone concentration, resulting in many paths not being found and the algorithm tends to converge prematurely. The global optimality of the solution is reduced.

Through simulation experiments and analysis the lost ants number of the traditional ant colony algorithm and the improved ant colony algorithm in this paper, explore the comprehensive advantages and disadvantages of the algorithm. Figure 6(a) shows the distribution of the lost ants number in each iteration of the algorithm, and Figure 6(b) shows the change of the lost ants number accumulated after each iteration of the algorithm.



(a) Iterative distribution of the lost ants number (b) Graph of the total number of iterations of lost ants

Figure 6. Graph of the lost ants number

Table 4. Table of the lost ants number

Algorithm	Lost ants number	Start and finish points	
		(3.6,25.2) (32.4,10.3)	(3.6,25.2) (4.4,8.3)
Ant colony algorithm	Maximum value	2381	2798
	Minimum value	2165	2662
	Average value	2273	2717.6
The improved ant colony algorithm in this paper	Maximum value	366	505
	Minimum value	259	389
	Average value	310.6	427.1

Table 4 shows the comparison table of the results obtained from the traditional ant colony algorithm and the improved ant colony algorithm in this paper for (3.6, 25.2)→(32.4, 10.3), (3.6, 25.2)→(4.4, 8.3) each run 10 times. M indicates the lost ants number. The comparison results indicate that.

(1) The improved ant colony algorithm in this paper can effectively reduce the lost ants number. In the simulation experiment of (3.6, 25.2)→(32.4, 10.3), the the minimum of the total lost ants number in the traditional ant colony algorithm is 2662, the maximum is 2798, and the average is 2717.6 a, compared with 389 at the minimum, 505 at the maximum, and 427.1 at the average of the improved algorithm. In the paths of (3.6, 25.2)→(32.4, 10.3) and (3.6, 25.2)→(4.4, 8.3), the average lost ants number of the improved algorithm is reduced by more than 90%, which are 92.68% and 93.64% respectively.

(2) The stability of the improved algorithm in this paper is high. Figure 6(a) shows that the traditional ant colony algorithm is easy to fall into local optimum, and the lost ants number is highly volatile throughout the iterations, and does not show a trend of decreasing generation by generation. In contrast, the lost ants number fluctuates steadily throughout the iterations, and shows a trend of decreasing generation by generation, and decreases to 0 in the 9th iteration.

DISCUSSION

In general, the performance of the improved algorithm in this paper is the best, and the number of iterations is slightly larger than that of the literature [11] algorithm, but the algorithm in this paper is more concise and faster in execution. However, from the overall comprehensive performance, the improved algorithm in this paper is relatively better in terms of the number of convergence iterations and running time, and has the best performance in terms of path length, number of turns, and turning angle, which indicates that the algorithm has higher performance stability, and the improved algorithm has advantages when searching for different paths, which meets the demand of emergency evacuation of actual cruise ship personnel.

REFRENCES

[1] Guo B, Kuang Z, Guan J, et al. An Improved A-Star Algorithm for Complete Coverage Path Planning of Unmanned Ships. International Journal of Pattern Recognition and Artificial Intelligence, 2022, 36(03): 2259009.

[2] Zheng D Y, Xia Y F, Teng H H, et al. Application of genetic algorithm to enhance the predictive stability of BP-ANN constitutive model for GH4169 superalloy. Journal of Central South University, 2024, 31(3): 693-708.

[3] Yan H, Zhong C Q, Wu Y H, et al. A hybrid-model optimization algorithm based on the Gaussian process and particle swarm optimization for mixed-variable CNN hyperparameter automatic search. Frontiers of Information Technology & Electronic Engineering, 2023(11): 1557-1574.

[4] Tang Y Y, Zheng E H, Qiu X. 3D UAV Trajectory Planning Based on Optimized Bidirectional A* and Artificial Potential Field Method. Journal of Air Force Engineering University, 2024, 25(5): 69-75.

[5] Marti K, Qu S. Adaptive stochastic path planning for robots–real-time optimization by means of neural networks[M]//Systems modelling and optimization. Routledge, 2022: 486-494.

[6] Low E S, Ong P, Low C Y, et al. Modified Q-learning with distance metric and virtual target on path planning of mobile robot. Expert Systems with Applications, 2022, 199: 117191.

[7] Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1996, 26(1): 29-41.
[8] Wang Y, Gao D, Wang X. Shortest path planning based on improved ant colony algorithm. ASP Transactions on Computers, 2021, 1(3): 6-11.
[9] Li S, Zhao G, Yue W. Research on Path Planning for Mobile Robot Based on Improved Ant Colony Algorithm[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 2026(1): 012049.
[10] Zhao H, Zhang C. An ant colony optimization algorithm with evolutionary experience-guided pheromone updating strategies for multi-objective optimization. Expert Systems with Applications, 2022, 201: 117151.
[11] Li L, Li H, Shan N. Path Planning Based on Improved Ant Colony Algorithm with Multiple Inspired Factor. Computer Engineering and Applications, 2019, 55(05): 219-225, 250.
[12] Zhang Y, Li Kui, Huang Chao. Path planning method for two-dimensional code mobile robot based on improved ant colony algorithm. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2021, 33(03): 491-497.
[13] Chen, Li-Qi. UAV Path Planning Based on Improved Ant Colony Algorithm. Computer and Digital Engineering, 2022, 50(03): 538-543.
[14] Shu Weinan, Zhao Jiansen, Xie Zongxuan, Zhang Student, Ma Xin. Path planning for unmanned surface vessels based on improved A* algorithm. Journal of Shanghai Maritime University, 2022, 43(02): 1-6.