Hindawi
Journal of Robotics
Volume 2022, Article ID 9229155, 11 pages
https://doi.org/10.1155/2022/9229155

Research Article

Robot Path Planning of Improved Adaptive Ant Colony System Algorithm Based on Dijkstra

Chonglin Gu, Ansong Feng, Guozhan Wang, and Xiqing Liu

Shenyang University of Chemical Technology, College of Information Engineering, Shenyang 110142, China

Correspondence should be addressed to Ansong Feng; 1034609859@qq.com

Received 9 September 2022; Revised 17 November 2022; Accepted 30 November 2022; Published 15 December 2022

Academic Editor: Keigo Watanabe

Path planning is one of the key technologies of robot. Aiming at the problems of slow convergence speed and inefficient search of traditional Ant Colony System Algorithm, an adaptive Ant Colony System Algorithm based on Dijkstra is proposed in the paper. Firstly, Dijkstra algorithm is applied to searching the initial path in the grid environment, constructing the initial path, optimizing the initial pheromone in the region, therefore, the Ant Colony System Algorithm avoid falling into blind search in the initial stage; In the transition probability, the disguised angle probability function and parameter adaptive pseudo-random proportion rule are introduced to improve the search efficiency and convergence speed of the algorithm, and eliminate the inferior ant path; Finally, B-spline interpolation curve is used to smooth the path. Compared with the traditional Ant Colony System Algorithm, the simulation results in the grid environment demonstrating its effectiveness to improve convergence speed and to enhance search efficiency are provided. The characteristics of the improved Ant Colony System Algorithm are faster convergence speed and better planning.

1. Introduction

Path planning is one of key factors in many research directions of intelligent robots. It is an indispensable part of robots to realize autonomous movement and navigation [1], path planning has applications in various fields, such as self-driving cars, medical applications, graphical animation, cell transportation, and robotic surgery [2–6]. The purpose of path planning is to plan the path with the optimal comprehensive index according to the position of the starting point, the target point and the environment of the robot [7, 8]. At present, traditional algorithms have been applied to robot path planning, such as Dijkstra algorithm [9] and A$*$ algorithm [10]. As the application scenarios of robots become more and more extensive, the comprehensive performance index requirements of robot planning paths are also getting higher and higher. Swarm intelligence optimization algorithms such as Artificial Fish Swarms Algorithm (AFSA) [11], the Ant-Colony optimization [12], Particle Swarm Optimization (PSO) [13] have been applied to solving robot path planning problems.

Ant colony optimization (ACO) is a swarm intelligence optimization algorithm based on the way of ants looking for food. The positive feedback mechanism and good robustness of the Ant-Colony Algorithm make it widely used to solve robot path planning problems, but it also has some defects, such as deadlock, slow search speed and local convergence. Therefore, many scholars have proposed corresponding improvement methods. The concept of elitist ant system (EAS) is proposed in literature [14]. On the basis of basic pheromone updating, the pheromone increment on the optimal path is applied to improving the convergence speed. However, with the increase of elite ants, the pheromone on the optimal path increases rapidly, which reduces the population diversity [15]. Bullnheimer B et al. Proposed the rank-based ant system (RAS), which only updates a certain number of pheromones on the better ant path [16]. The ACS algorithm is an extension of the ant colony optimization algorithm. The ACO algorithm can generate the optimal path from the previous suboptimal path information [17, 18]. Ant colony system (ACS) algorithm is one of the most excellent ant colony optimization algorithms.

Compared with the first ant colony optimization algorithm, ant system (AS) algorithm, its performance has been greatly improved.

Compared with the existing relevant studies [19–22], although certain effects have been achieved in avoiding falling into the local optimization problem and improving the convergence speed of ant colony algorithm, there are still problems such as poor global path effect and long searched path length. In order to better solve these problems and obtain better quality paths The contribution of this paper is to propose a new path planning method based on Dijkstra's improved adaptive ACS algorithm. In this paper, the following improvements are made based on the traditional ant colony algorithm: Searching the initial path in the grid environment through Dijkstra algorithm, expanding centred on the path, and optimizing the initial pheromone concentration in the extended area; The combination of steering cost and adaptive pseudo-random proportional rule is introduced to improve the path quality and the pheromone updating strategy is improved. The adaptive weights of high-quality ants are updated, and inferior ants are not updated to avoid interference. In addition, B-spline curve is employed to smooth and optimize the path.

The novelty of this algorithm is that it improves the performance of the ACS algorithm and to reach the optimal value with better convergence speed. After each iteration, the last pheromone will be evaluated to improve the quality of the final path. These evaluation results will affect the results of the next iteration. Then, the performance of the proposed algorithm is compared with the basic ACS, RAS and EAS algorithms using several benchmark cases. The improved algorithm is compared with other existing algorithms, and the results show that the improved algorithm has better convergence speed. The proposed algorithm reached those optimal value based on the test results from some benchmark cases with known optimal values. Statistical analysis shows that the improved algorithm is significantly different from other existing algorithms in performance, and the improved algorithm has good convergence speed.

2. Related words

2.1. Basic ant colony system algorithm. According to the phenomenon of ant colony foraging in nature, Dorigo and Gambardella first proposed the ant colony system (ACS) algorithm and applied it to solving the TSP problem. When compared to other optimization algorithms, the ACS algorithm still shows its superiority. Dorigo and Gambardella reported that the ACS algorithm could produce a 2.2% better path length than the genetic algorithm (GA), 1.3% better than evolutionary programming (EP), and 7.8% better than simulated annealing (SA) [23, 24]. Because the ACS algorithm is better than the ACO in terms of the cost of the route generated, and the computational time required in literature [25], Therefore, this paper uses ACS algorithm as the basis for improvement. In ACS algorithm the foraging behaviour of ants in nature is simulated, and certain intelligence is added to ants which is mainly achieved by adding a certain memory to the ants and processing the unique pheromone of ants-pheromone volatilization. Therefore, the performance of the algorithm is further improved [26]. At the beginning, since no ants pass through the path and no pheromone, ants randomly choose the path they encounter. With the passage of time, more and more ants will pass through the path with short distance and the pheromone concentration accumulated on the path will be higher and higher. Therefore, the probability of subsequent ants choosing this path will also increase. Finally, the ant colony will find an optimal path to the target point.

2.2. Mathematical model of ant colony system algorithm

2.2.1. Formula of transfer probability. The probability that the ants in the ant colony choose the next node to move according to the pheromone concentration. It is also affected by the expected information between the current node and the surrounding nodes [27]. The state transition rule for ants is shown in (1):

$$p_{ij}^{k} = \left\{ \frac{\left[\tau_{ij}\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{k\in d_k} \left[\tau_{ij}\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}, \text{if } j \in d_k, 0, \text{otherwise}} \right. . \quad (1)$$

P_{ij}^{k} is the transition probability of ant k selecting node j at node i, where τ_{ij} is the pheromone concentration left by ants on path (i, j). η_{ij} is the heuristic information between i and j, which is generally taken as $\eta_{ij} = 1/d_{ij}$, and it represents the Euclidean distance between nodes i and j. α is the information heuristic factor, which indicates the relative importance of pheromone concentration when ants select the next node; β is the expected heuristic factor and indicates the importance of the distance between the current node and the next node. The nodes passed by ants are recorded in the $tabu_k$ (k = 1, 2, ... M, M is the number of ants). $d_k = \{C - \text{tabu}_k\}$ is the set of nodes that ants can choose, and C is the set of all nodes. The decision rule of ants moving from the current node to the next node is determined by the pseudo-random proportional rule. The pseudo-random proportional rule equation is shown in equation (2) [28];

$$\text{P} = \begin{cases} \max\left\{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}\right\}, & q \leqslant q_0, \\ P_{ij}^{k}(t), & q > q_0, \end{cases} \quad (2)$$

where: P is selected as the next grid position; q is a random number range from 0 to 1; q_0 is a coefficient.

2.2.2. Pheromone concentration update rule. Tabu table named tabu_k is used to record the nodes passed by ants in the current iteration until this iteration is completed, otherwise the nodes recorded in the tabu table cannot be accessed again. When all ants complete the access to all nodes, in the tabu table, the nodes passed by each ant are recorded to form the feasible solution of the path. After that, the previous path pheromone should be weakened, and the newly introduced pheromone should be strengthened. When the ant completes a path search, it will leave pheromone on the path, and part of the previous pheromone will volatilize. The pheromone concentration at the next time is shown in (3) and (4) [29]:

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

$$\Delta\tau_{ij}^{k}(t,t+1) = \begin{cases} \dfrac{Q}{L_k}, & \text{Path of ant } k\,(i),(j), \\ 0, & \text{others.} \end{cases} \tag{4}$$

$\tau_{ij}(t+1)$ represents the pheromone concentration on the updated (i, j) path, ρ is pheromone volatilization factor, $\tau_{ij}(t)$ is the amount of pheromone increased on the path (i, j) after the last iteration, $\Delta\tau_{ij}^{k}(t,t+1)$ is the pheromone quantity that is left by ant k on (i, j) in this cycle, L_k is the path length visited by ant k in this iteration, Q is a constant that is set to 1, and it is the pheromone enhancement coefficient.

2.3. Dijkstra algorithm. Dijkstra algorithm is employed to calculate the shortest path from one vertex to other vertices, which is applied to solving the shortest path problem in weighted graph [30]. According to the principle of array, Dijkstra algorithm can quickly adapt to the update of conditional variables and achieve the purpose of rapid solution with the increase of calculation layers. Dijkstra algorithm adopts the strategy of greedy algorithm, and each time it traverses the adjacent nodes of the vertex that is closest to the starting point and has not been visited, until it extends to the end point [31]. The process of Dijkstra algorithm path planning is shown in figure 1.

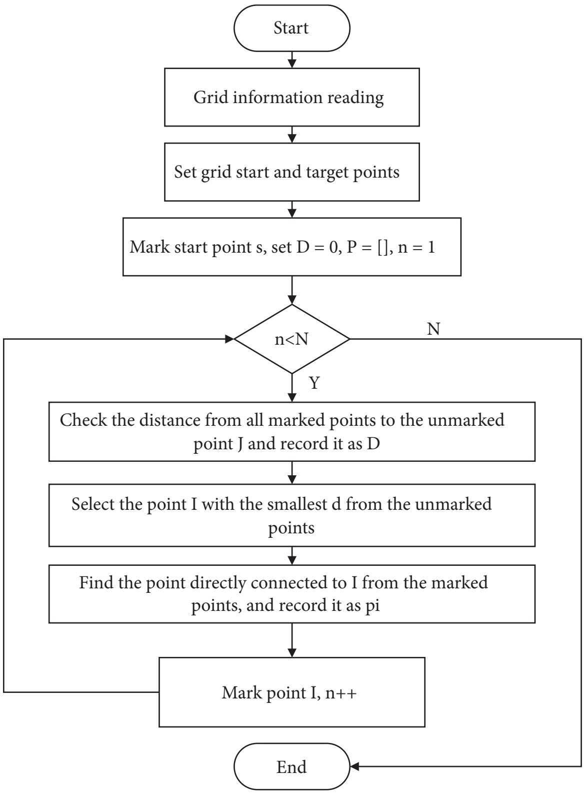
Start
Grid information reading
Set grid start and target points
Mark start point s, set D = 0, P = [], n = 1
n<N
N
Y
Check the distance from all marked points to the unmarked point J and record it as D
Select the point I with the smallest d from the unmarked points
Find the point directly connected to I from the marked points, and record it as pi
Mark point I, n++
End

Figure 1: Flow chart of Dijkstra algorithm path planning

3. Improved adaptive ant colony system algorithm based on Dijkstra

3.1. Improvement of initial pheromone. In the traditional Ant Colony System Algorithm, the initial pheromone concentration is evenly distributed, which leads to the blind search in the early stage and reduces the search speed. In this paper, Dijkstra is employed to improve the planning of the initial path, and then each grid of the initial path is expanded to 8 directions to construct the dominant area for pheromone optimization. The number of expanded grids is related to the size of the grid map and the proportion of obstacles in the grid map. In this way, the algorithm can not only avoid the blind search in the early stage, but also do not completely rely on the guidance of the initial path. The grid forming the initial path expands to the adjacent grids in 8 directions. The specific pheromone diffusion concentration method is shown in Figure 2. The yellow squares represent the initial starting position of the ants, the purple squares represent the target end position of the ants, the green squares are the initial paths planned by the Dijkstra strategy, and the red mesh area is the initial pheromone diffusion area planned by the Dijkstra strategy.

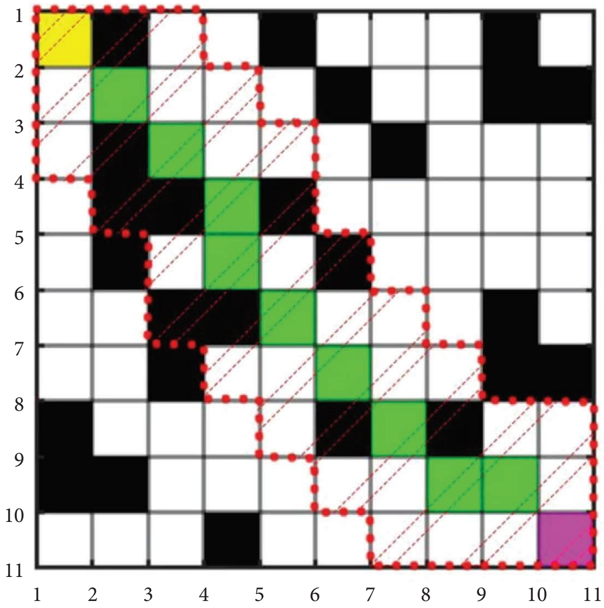

Figure 2: Dijkstra algorithm expands the grid pheromone concentration map

The initial pheromone of the feasible grid in the grid area is optimized. The red reticulated areas were given relatively high pheromone concentrations relative to other areas, which is expressed as (5) and (6):

$$\tau_{ij}=\begin{cases} u*\tau_0, & i,j\in U,\\ \tau_0, & \text{others},\end{cases} \tag{5}$$

$$\tau_0=\frac{1}{((X+Y)*(X*Y))}. \tag{6}$$

The τ_0 is the initial pheromone concentration; u is the enhancement coefficient of pheromone concentration; U represents the adjacent grid range planned by Dijkstra algorithm; X is the number of rows of grid map, and Y is the number of columns of grid map.

3.2. Heuristic information. The traditional distance heuristic information has weak inspiration for ACS algorithm, resulting in low search efficiency. Therefore, the reciprocal of the length from the optional node to the end point is used as the distance heuristic method [32], that is, the heuristic information from the current node to the next node is the Euclidean distance from the next node to the target node. Heuristic information is generated according to the following (7):

$$d_{jg}=\text{distance}(j,\text{goal}). \tag{7}$$

3.3. Formula of transition probability. The traditional ACS algorithm adopts the roulette method in path selection, without considering the back and forth of ants that are looking for the path. Therefore, in the early stage of the search path or the early stage of the iteration, the path is too long, which affects the speed of the algorithm in the later iteration. The improved ACS Algorithm introduces a strategy of deflection direction probability, and Figure 3 illustrates the way of direction change when the ant colony algorithm performs grid selection. As shown in Fig. 2, the directional change in the grid diagram is divided into types. The θ angles are 0、 $\pi/4$、 $\pi/2$、 $3\pi/4$ and π in eight cases. According to the magnitude of θ, the corresponding probability cost of deflection direction can be calculated. Each time the ant moves to the next grid, it determines the probability of deflection of all eight adjacent moveable grids. The main purpose of this is to constrain the path from taking extreme turns and folding back to affect the length of the path planning algorithm.

The deflection direction probability is introduced to the grid selection probability of the traditional ACS Algorithm.

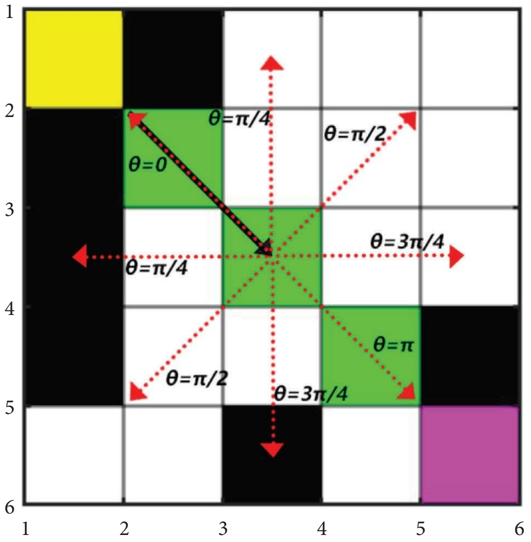

Figure 3: Illustration of direction change mode generated by transition probability

The transfer probability equation of the improved ACS algorithm is:

$$p_{ij}^{k}=\begin{cases}\dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}[\zeta_{ij}]^{\gamma}}{\sum\limits_{j\in\text{allowed}_k}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}[\zeta_{ij}]^{\gamma}}, & \text{if } j\in d_k,\\ 0,\ \text{otherwise},\ \zeta_{ij}=\dfrac{\theta}{\pi}. & \end{cases} \tag{8}$$

The ζ_{ij} is the steering cost from the current node to the next feasible node; γ is the steering coefficient, which determines the influence of steering cost in the calculation of transition probability; θ is the deflection direction angle generated by the ant when selecting the grid.

3.4. Adaptive strategy of probability transfer. The selection of the next grid in the path adopts the pseudo-random proportion rule varying with the number of iterations in roulette, in which q_0 is adaptive probability, and the value is improved as following:

$$\begin{gathered} p=\begin{cases}\max\left\{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}[\zeta_{ij}]^{\gamma}\right\}, & q\le q_0,\\ P_{ij}^{k}, & q>q_0,\end{cases}\\ q_0=u*\left(\frac{d_{sg}}{L_{\text{best}}}\right)*e^{-NC/NC_{\max}},\\ d_{sg}=\text{distance}(\text{start},\text{goal}). \end{gathered} \tag{9}$$

where: P is selected as the next grid position; q is a random number range from 0 to 1; $NC_{\max}$ is the maximum value of iteration; NC is the number of current iterations; u is a coefficient and the value is 1.1, that is used to adjust the range of q_0; d_{sg} is the distance from the starting point to the target point; L_{best} is the optimal path length in the current iteration times; In the early stage of the algorithm iteration, the ratio of $NC/NC_{\max}$ is small, so the value of q_0 is large, and the path with less number of direction changes and low cost of direction change can be selected, which can speed up the convergence of the algorithm. In the later stage of the algorithm iteration, as the number of iterations increases, the ratio of $NC/NC_{\max}$ increases gradually, q_0 decreases with the increase of the number of iterations, so that the algorithm obtains better adaptiveness and reduces the probability of falling into local convergence.

3.5. Smooth path planning curve. The path of the grid map is composed of several straight lines connecting the geometric center of the grid, so there must be spikes at the turning, that is inconsistent with the motion characteristics of the robot. The optimal path of the improved ACS Algorithm is still not smooth enough, and there are peak inflection points in some paths. In order to solve this problem, it is necessary to smooth the path to satisfy the requirements of real robots. The mathematical expression of B-spline curve is relatively simple. It has the ability of local modification that Bezier curve does not have, and it is second-order continuous at the connection [33, 34]. Therefore, it is widely used in image processing, the shape design, etc. The path is smoothed using a cubic B-spline, and its expression is:

$$P_{k,n}(t) = \sum_{i=0}^{n} P_i G_{i,n}(t), \quad t \in [0,1], \tag{10}$$

where P_i is the control vertex (i =0, 1, ..., n), i denotes the B-spline number; $G_{i,n}(t)$ the nth order B-sample basis function, which can be derived from DeBoor's recurrence formula as:

$$G_{i,n}(t) = \frac{1}{n!} \sum_{j=0}^{n-i} (-1)^j C_{n+1}^{j} (t+n-i-j)^n. \tag{11}$$

Let $n = 3$, then the basis function of the cubic B-spline curve can be deduced as:

$$\begin{cases} G_{0,3}(t) = \frac{1}{6}\left(-t^3 + 3t^2 - 3t + 1\right), \\ G_{1,3}(t) = \frac{1}{6}\left(3t^3 - 6t^2 + 4\right), \\ G_{2,3}(t) = \frac{1}{6}\left(-3t^3 + 3t^2 + 3t + 1\right), \\ G_{3,3}(t) = \frac{1}{6}t^3. \end{cases} \tag{12}$$

The equation of the cubic B- spline curve can be obtained from the above equation:

$$P_{0,3}(t) = \frac{1}{6}\begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}. \tag{13}$$

3.6. Improved algorithm flow

Step 1. establish a grid map. The path finding map is planned, a reasonable raster map is established, and the starting point S and the ending point G are set.

Step 2. set the parameter value. The number of ants in each iteration M, the maximum number of iterations $NC_{\max}$, pheromone evaporation factor ρ and other related parameter values of ant colony calculation are set.

Step 3. initial pheromone optimization. The Dijkstra algorithm is used to search the initial path in the grid map, and according to the size and complexity of the raster map, the initial pheromone enhancement region is obtained.

Step 4. select the path grid. Place the ant at the starting point, select the next grid according to the selection probability and roulette. Place the serial number of the passing grid in the taboo table and record the current walking length. When the ant reaches the end or falls into a deadlock, the selection ends.

Step 5. update the global pheromone. After all the ants in this iteration complete the path finding, the pheromone of the ants that travel to the end is updated.

Step 6. judge whether the maximum number of iterations is reached. If the number of iterations reaches the set maximum value, output the optimal result of routing and end the operation. If not, repeat step 4 and 5 until the maximum number of iterations is reached.

Step 7. smooth the optimal path. The peak of the optimal path is smoothed by cubic B-spline curve, and the smoothed path and length are output.

The diagram of the improved Ant Colony System Algorithm is shown in figure 4:

4. Experiments

In order to verify the feasibility of the improved Ant Colony Sytem Algorithm, the algorithm is placed in 20 × 20 and 50 × 50 these two grid environments with different sizes and complexity. The environment adopts Windows 10 home Chinese operating system, core (TM) i7-8565u processor, 8 GB running memory and MATLAB 2018b. In order to

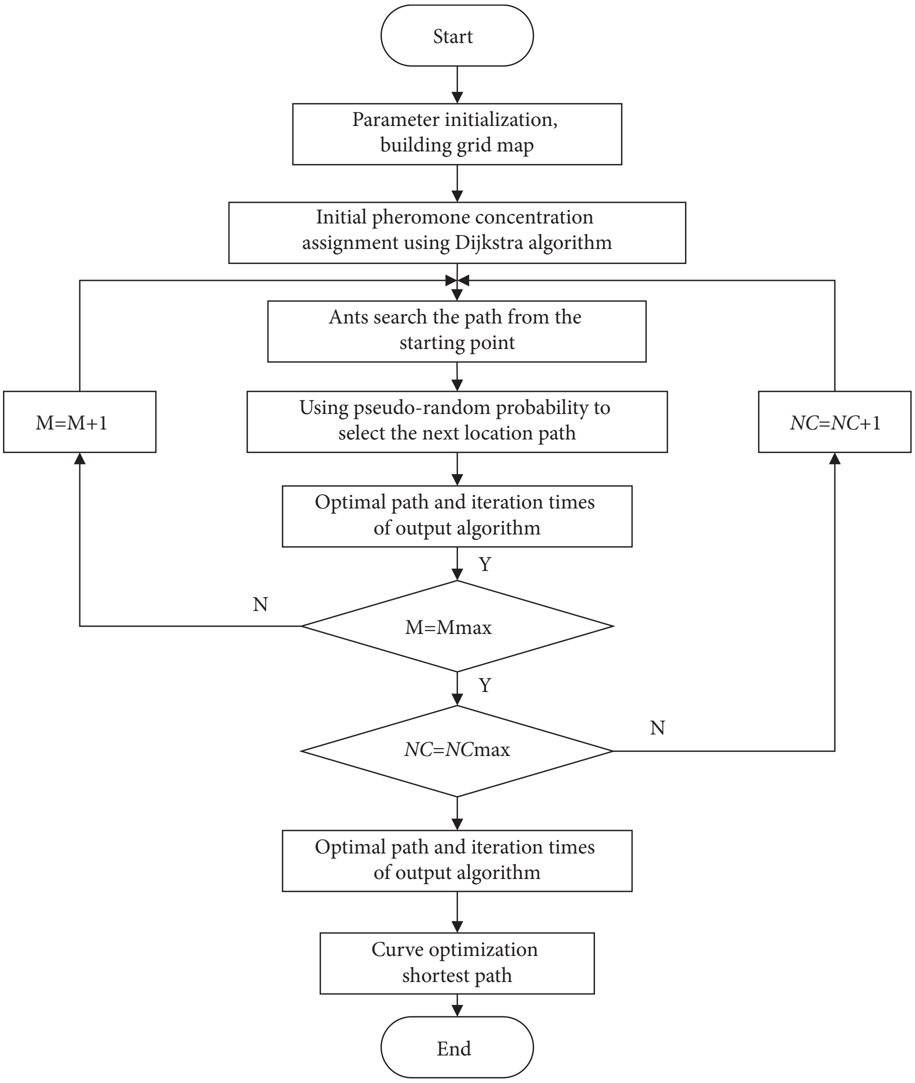
Start
Parameter initialization, building grid map
Initial pheromone concentration assignment using Dijkstra algorithm
Ants search the path from the starting point
Using pseudo-random probability to select the next location path
M=M+1
NC=NC+1
Optimal path and iteration times of output algorithm
Y
N
M=Mmax
Y
NC=NCmax
N
Optimal path and iteration times of output algorithm
Curve optimization shortest path
End

Figure 4: Flow chart of improved ant-colony system algorithm

eliminate the influence of map environment factors, two different obstacle simulation maps are established: one is a U-shaped obstacle environment map (specification 20 ∗ 20), and the other one is a complex obstacle environment map (specification 50 ∗ 50). The starting point of the two map environment models of path planning is 1, and the target point is 400 and 2500, respectively.

4.1. Environment establishment of 2D grid. Compared with visual graph method, the node method and free space method, the grid method has the advantages of consistent expression of planning space and convenient for computer modelling, storage, processing, updating and analysis [35]. Therefore, the grid method is used to verify the environment. Grid map is applied to describing the characteristic information of the real environment without too much time and space. At the same time, grid information is employed to mark the starting point, ending point, obstacles and other complex environmental features. When dealing with irregular obstacles, grid representation has obvious advantages over other map description methods [36]. Therefore, this paper uses the grid method to establish the working environment model of mobile robot. According to the obstacle distribution in the environment and the above grid division rules, the grid map includes feasible grid and the obstacle grid. An example of a two-dimensional grid environment is shown in figure 5, in which the white grid represents the feasible grid and the black grid represents the obstacle grid.

4.2. U-shaped simulation environment. 10 experiments are carried out for each algorithm, and then the average of the best values of 10 experiments are averaged to initialize the information. The initialization information sets the number of ants in each generation M=30, the maximum number of iterations $NC_{\max}$ = 50, and the initial pheromone concentration α =1, β =5, γ =0.1, ρ =0.08, pheromone concentration Q =1. In the simple environment map model, simulation results of optimal path planning based on the basic Ant Colony System Algorithm and the improved Ant Colony

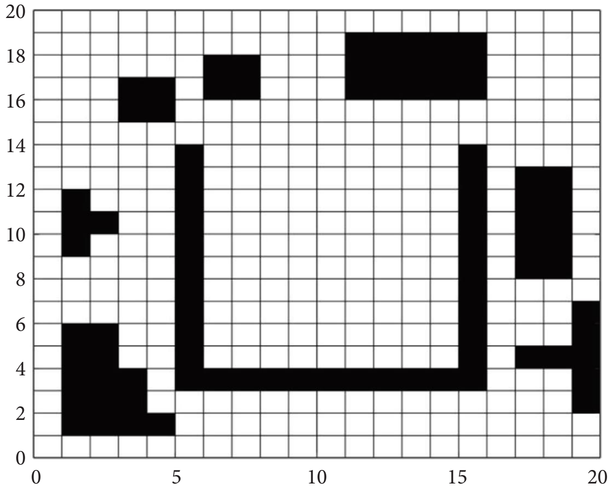

Figure 5: 2D grid environment diagram

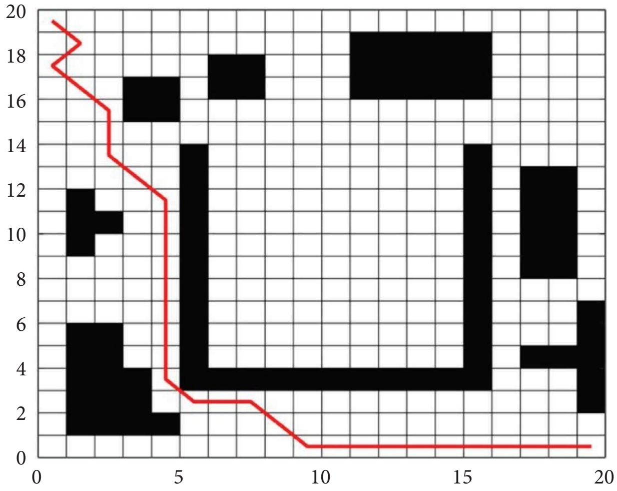

Figure 6: Basic ant colony system algorithm path planning

System Algorithm are shown in Figure 6 and Figure 7, and their iterative convergence curves are shown in figure 8 and figure 9.

In the U-shaped region, the traditional ACS Algorithm has the potential for the ants to get stuck in the U-shaped region leading to difficulties in obtaining the minimum path after several iterations, affecting the optimal length of the final path planning algorithm. Then in this experiment a large U-shaped obstacle was placed in the center of the map. Figure 6 and Figure 7 show the optimal path planning diagrams obtained from the two algorithms within ten experiments. Figure 8 shows a comparison of the algorithm path convergence curves for the U-shaped simulation environment. It can be seen in this experiment that the improved algorithm did not obtain the optimal value when it fell into the U-shaped region in the first iteration, but after another iteration the improved algorithm quickly got rid of the U-shaped obstacle to obtain the optimal path planning quickly. It can be seen from Figure 8 that the length of the optimal path planned by the improved ACS Algorithm is significantly less than the path distance planned by the basic ACS Algorithm. The convergence length of the improved algorithm in this paper is 33.8995, and it begins to converge at the first generation; The path convergence length of the basic ACS Algorithm is 34.7279 and the convergence algebra is 22 generations. Obviously, the improved ACS Algorithm is better than the traditional ACS Algorithm in the optimal path length and convergence speed.

4.3. Complex obstacle environment. 10 experiments are carried out for each algorithm, and then the average of the best values of 10 experiments are averaged to initialize the information. The initialization information sets the number of ants in each generation M=30, the maximum number of iterations $NC_{\max}$ = 50, and the initial pheromone concentration α =1, β =5, γ =0.1, ρ =0.08, pheromone concentration Q =1. In the simulation experiment of complex obstacle, elite Ant-Colony Algorithm and sorting Ant-Colony Algorithm are also introduced to compare the effect of the algorithm.

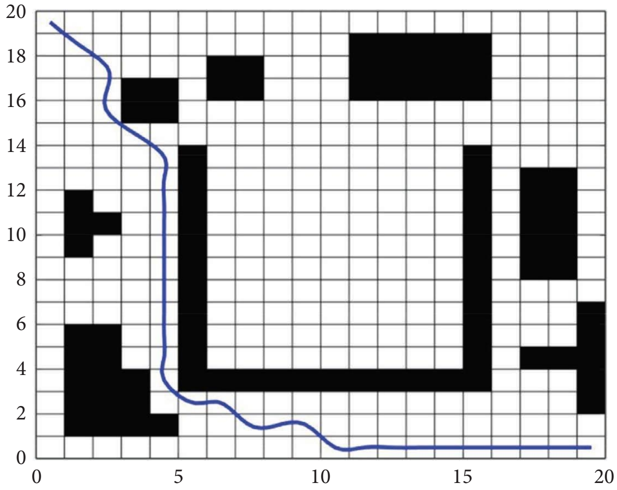

Figure 7: Improved ant colony system algorithm path planning

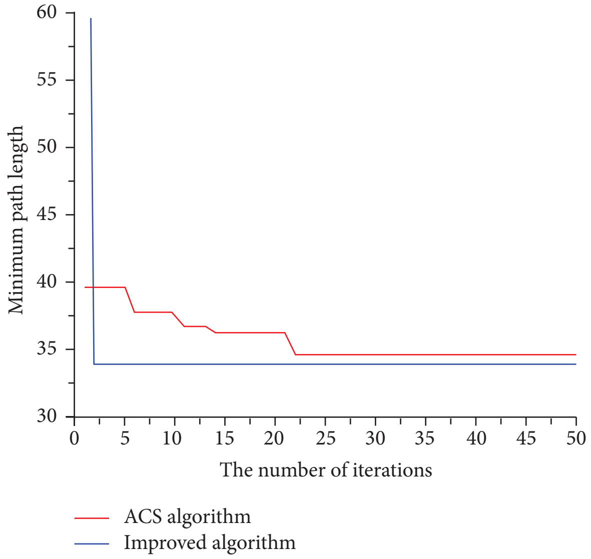

Figure 8: Comparison of algorithm path convergence curves for U-shaped simulation environment

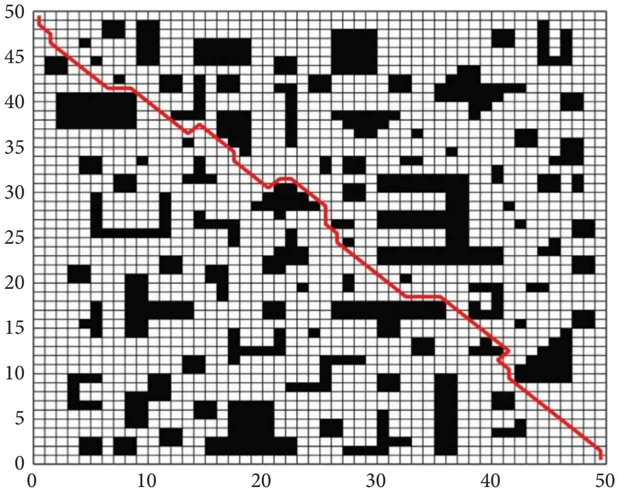

Figure 9: Basic ant colony system algorithm path planning

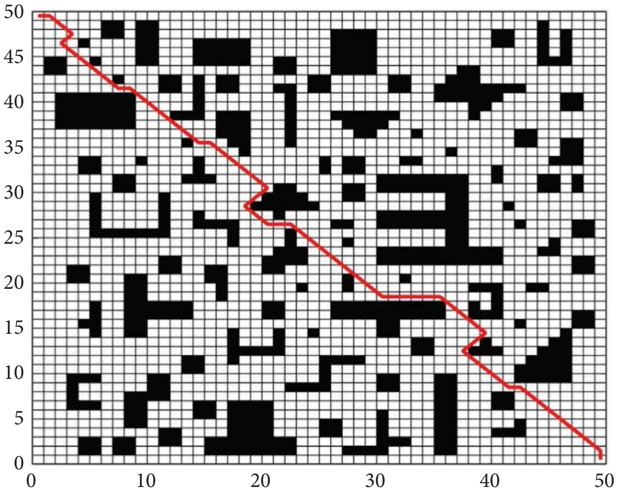

Figure 10: EAS algorithm path planning

The simulation results of several algorithms for optimal path planning under complex obstacle environment maps are shown in Figure 9 to Figure 10, Figure 11, Figure 12, along with their iterative convergence curves as shown in Figure 13.

According to the analysis of the above figures, the performance of the improved ACS Algorithm is better than that of the basic ACS Algorithm in the complex obstacle environment. The improved ACS Algorithm converges to the optimal path in generation 1, and the path length is 75.6396, while the optimal path length of the basic ACS Algorithm is 77.68, and the convergence algebra is 43. In the two map environments, it is shown in figures that in both map environments, the global optimization and convergence speed of the improved ACS Algorithm are better than that of the basic ant colony system algorithm by introducing a new judgment probability, strengthening the initial path pheromone concentration and adaptive adjustment. Therefore, the optimization efficiency of ants is improved.

In table 1, it is illustrated that compared with the basic ACS Algorithm, the optimal iterative convergence times of this algorithm are reduced by 66.7%, the inflection point is reduced by 34.8%, and the path length is reduced by 3.4%. Compared with EAS algorithm, the optimal convergence times of this algorithm are reduced by 83.3%, the inflection point is reduced by 16.7%, and the path length is reduced by 0.79%. Compared with RAS algorithm, the optimal convergence times of this algorithm are reduced by 85.7%, the inflection point is reduced by 28.6%, and the path length is shortened by 1.56%. According to the data of standard deviation, the stability of this algorithm is better than other algorithms in ten experiments. Therefore, the convergence speed and convergence results of the proposed algorithm in the 50×50 grid environment are better than those of the comparison algorithms.

The improved algorithm can also be well implemented in a larger environment with more obstacles, for example in a 100×100 grid map as shown in Figure 14, and the algorithm can be adapted well in larger grid environments.

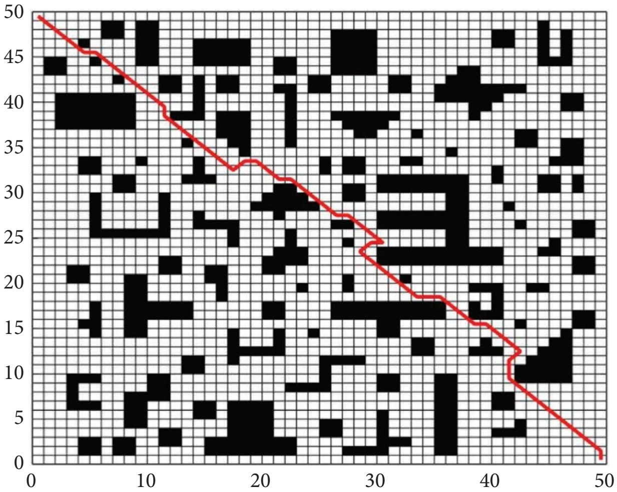

Figure 11: RAS algorithm path planning

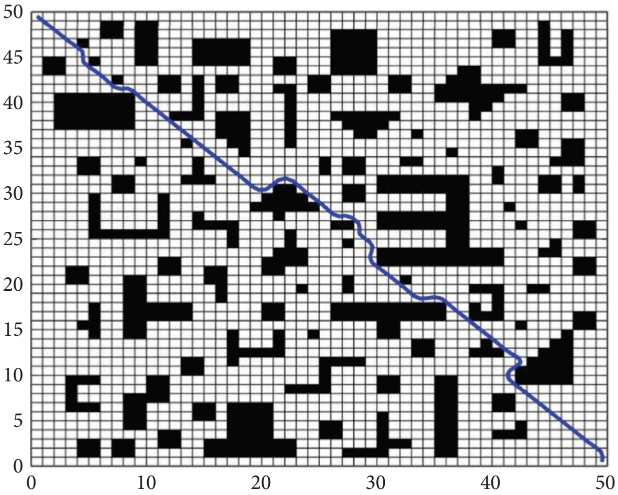

Figure 12: Improved ant colony system algorithm path planning

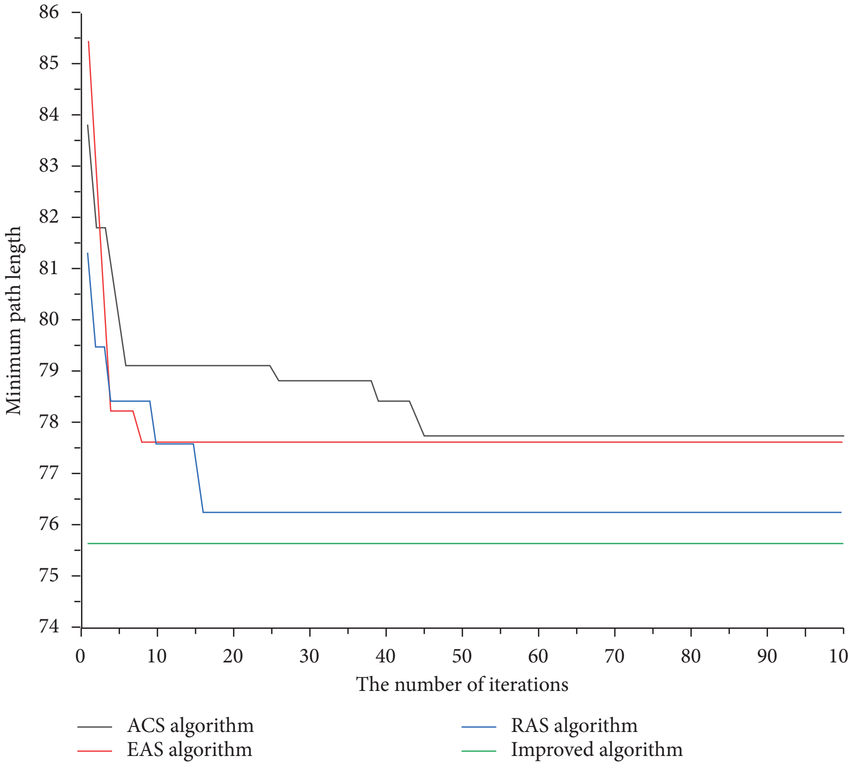

FIGURE 13: Comparison of algorithm path convergence curve in complex obstacle simulation environment

TABLE 1: Comparison tests of 50 × 50 simulation environment

Algorithm	Optimal path length	Worst path length	Iterations of optimal path	Number of inflection points	Standard deviation of data
Improved algorithm	75.6396	79.05	1	15	1.0342
ACS algorithm	77.68	81.8823	3	23	1.5052
EAS algorithm	77.64	80.2254	6	18	1.6050
RAS algorithm	76.2254	83.8823	7	21	2.2678

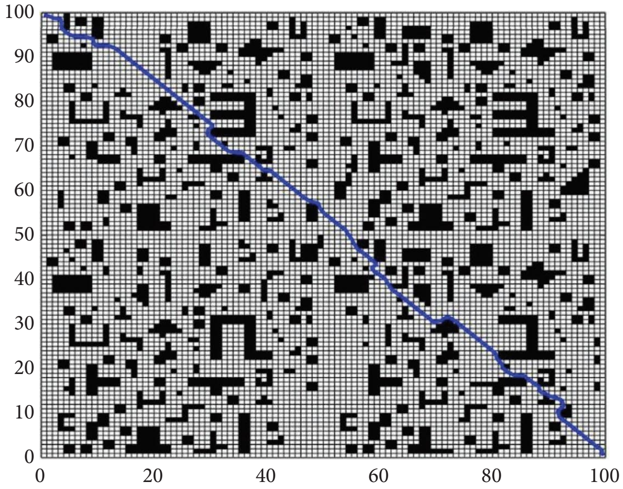

FIGURE 14: Path planning of improved algorithm for large-scale complex obstacle simulation environment

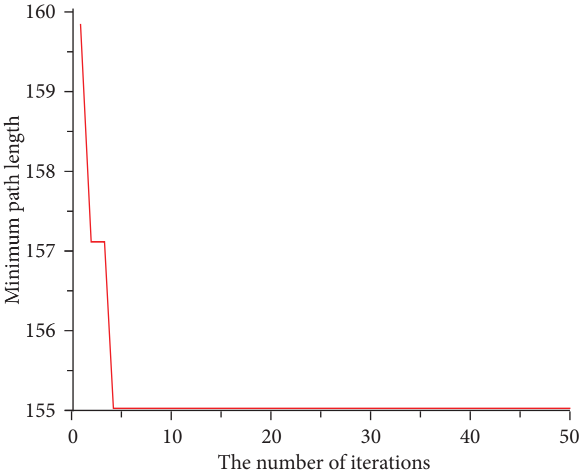

FIGURE 15: Algorithm path convergence curve for large scale complex obstacle simulation environment

Figure 15 shows the convergence curve of the optimal path after ten experiments. It can be seen that the improved algorithm also converges rapidly in large-scale and complex obstacle environments. The best path planning route is basically found within five iterations in all ten experiments, proving that the algorithm is also well adapted to complex environments and finds the optimal path planning route.

5. Conclusions

According to the size of grid environment and the proportion of obstacle grid, the algorithm in the paper that is combined with the path searched by Dijkstra algorithm optimizes the initial pheromone of Ant Colony System Algorithm through grid expansion, which effectively improves the search speed of the algorithm in the early stage. The number of inflection points in search path are reduced and the convergence of the algorithm is promoted by introducing the variable heuristic function and the adaptive pseudo-random proportion rule. The cubic spline is used to smooth the spikes generated by the path in the grid environment, which effectively reduces the generation of sharp paths. Comparing results in the grid environment, it is shown that the path searched by the improved adaptive Ant Colony System Algorithm based on Dijkstra algorithm is better than that of others, and the optimal path is obtained with less iteration. In this paper, as a global path planning algorithm for mobile robots, the improved Ant Colony System Algorithm is mainly applied to the static environment where the global map is known. In this paper, as a global path planning algorithm for mobile robots, the improved Ant Colony System Algorithm is mainly applied to the static environment where the global map is known. For the research in dynamic environment, the adaptive Ant Colony System Algorithm based on Dijkstra algorithm will be considered in combination with other algorithms, or the research of three-dimensional path planning will be carried out.

Data Availability

The data used to support the findings of the study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by Foundation of the Natural Science Foundation of Liaoning Province (2018055012), Foundation of Liaoning Province Education Administration (LJKZ0448)

References

[1] W. Zhe and L. Fang, "Self-adaptive ant colony algorithm based on statistical analysis and its application," in *Proceedings of the 2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pp. 313–317, Shenyang, China, July 2019.

[2] L. Yu, Z. Wei, H. Wang, Y. Ding, and Z. Wang, "Path planning for mobile robot based on fast convergence ant colony algorithm," in *Proceedings of the 2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1493–1497, Takamatsu, Japan, August 2017.

[3] Y. Gong, W. Wang, and S. Gong, "Research of a self-adaptive mixed-variable multi-objective ant colony optimization algorithm," in *Proceedings of the 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (Smart-Data) and IEEE Congress on Cybermatics (Cybermatics)*, pp. 735–742, Rhodes, Greece, November 2020.

[4] X. Zheng, Y. Li, and H. Liu, "A study on path planning of crowd animation based on pso," in *Proceedings of the 2010 Sixth International Conference on Natural Computation*, pp. 2566–2570, Yantai, China, August 2010.

[5] T. Ju, S. Liu, J. Yang, and D. Sun, "Apply rrt-based path planning to robotic manipulation of biological cells with optical tweezer," in *Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation*, pp. 221–226, Beijing, China, August 2011.

[6] R. C. Jackson and M. C. Çavuşoğlu, "Needle path planning for autonomous robotic surgical suturing," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, pp. 1669–1675, Karlsruhe, Germany, December 2013.

[7] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, April 2016.

[8] S. Campbell, N. O'Mahony, A. Carvalho, L. Krpalkova, D. Riordan, and J. Walsh, "Path planning techniques for mobile robots a review," in *Proceedings of the 2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, pp. 12–16, Barcelona, Spain, February 2020.

[9] S. Julius Fusic, P. Ramkumar, and K. Hariharan, "Path planning of robot using modified dijkstra Algorithm," in *Proceedings of the 2018 National Power Engineering Conference (NPEC)*, pp. 1–5, Madurai, India, March 2018.

[10] Z. Zhang, S. Wang, and J. Zhou, "A-star algorithm for expanding the number of search directions in path planning," in *Proceedings of the 2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pp. 208–211, Shanghai, China, October 2021.

[11] Y. Huang, P. Wang, M. Yuan, and M. Jiang, "Path planning of mobile robots based on logarithmic function adaptive artificial fish swarm algorithm," in *Proceedings of the 2017 36th Chinese Control Conference (CCC)*, pp. 4819–4823, Dalian, China, July 2017.

[12] Y. Zhang, X. Liu, and Y. Zhang, "Scheduling of multiple mobile robots based on time windows and improved ant colony algorithm," in *Proceedings of the 2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, pp. 77–81, Chongqing, China, November 2020.

[13] Y. Gong, M. Luo, C. Wang, and T. Han, "A path planning method based on improved particle swarm optimization algorithm," in *Proceedings of the 2020 7th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 106–110, Changsha, China, December 2020.

[14] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[15] A. Verma and P. Tripathi, "Rank based ant colony optimization for energy efficient VM placement on cloud," in *Proceedings of the 2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 1020–1026, Coimbatore, India, January 2018.

[16] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A new rank based version of the ant system - a computational study," *Central European Journal of Operations Research*, vol. 7, no. 1, pp. 25–38, 1999.

[17] A. Yang, L. Gao, and Y. Luo, "An improved ant colony system algorithm for optimal path planning problem of mobile robots," in *Proceedings of the 2010 Second International Conference on Computer Modeling and Simulation*, pp. 526–530, Sanya, China, January 2010.

[18] S. K. Çalık, "UAV path planning with multiagent Ant Colony system approach," in *Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU)*, pp. 1409–1412, Zonguldak, Turkey, May 2016.

[19] H. Zhao, "Optimal path planning for robot based on ant colony algorithm," in *Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 671–675, Limassol, Cyprus, June 2020.

[20] D. Ruan and M. jian, "Path planning of restaurant delivery robot based on improved ant colony algorithm," in *Proceedings of the 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, pp. 415–418, Xiamen, China , Xiamen, China, August 2019.

[21] Y. Xue, Y. Chen, Z. Ding, X. Huang, and D. Xi, "Robot path planning based on improved ant colony algorithm," in *Proceedings of the 2021 Power System and Green Energy Conference (PSGEC)*, pp. 129–133, Shanghai, China, July 2021.

[22] H. Liu, Y. Bao, L. Shao, and J. Li, "Robot path planning based on adaptive parameter ant colony algorithm," in *Proceedings of the 2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 710–714, Guilin, China, August 2022.

[23] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, April 1997.

[24] L. M. Gambardella and M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies," in *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 622–627, Nagoya, Japan, May 1996.

[25] R. Jangra and R. Kait, "Analysis and comparison among ant system; ant colony system and max-min ant system with different parameters setting," in *Proceedings of the 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, pp. 1–4, Ghaziabad, India, February 2017.

[26] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: an autocatalytic optimizing process," *Clustering*, vol. 3, no. 12, p. 340, 1991.

[27] H. Wang, L. Zhu, and J. Wang, "Research on path planing of parking system based on Dijkstra-Ant colony hybrid algorithm," *Chinese Journal of Engineering Design*, vol. 23, no. 5, pp. 489–496, 2016.

[28] S. Zhang, J. Pu, and Y. Si, "An adaptive improved ant colony system based on population information entropy for path planning of mobile robot," *IEEE Access*, vol. 9, pp. 24933–24945, 2021.

[29] D. Zhang, X. You, S. Liu, and H. Pan, "Dynamic multi-role adaptive collaborative ant colony optimization for robot path planning," *IEEE Access*, vol. 8, pp. 129958–129974, 2020.

[30] H. I. Kang, B. Lee, and K. Kim, "Path planning algorithm using the particle swarm optimization and the improved dijkstra algorithm," in *Proceedings of the 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, pp. 1002–1004, Wuhan, China, December 2008.

[31] Z. Liu, Y. Wang, and P. Zhou, "Research on dynamic scheduling algorithm of aviation weapon automatic production line based on dijkstra algorithm," *Aero Weaponry*, vol. 26, no. 3, pp. 94–98, 2019.

[32] X. Liu, G. Zeng, Bo Huang, and D. Lu, "Research on path planning of mobile robot based on improved ant colony algorithm," *Electronic Science and Technology*, vol. 32, no. 9, pp. 5–925, 2019.

[33] X. Wu, B. Guo, and J. Wang, "Mobile robot path planning algorithm based on particle swarm optimization of cubic splines," *Robot*, vol. 31, no. 6, p. 5, 2009.

[34] Y. Zehua, "MTSP UAV detection problem based on ant algorithm," in *Proceedings of the 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pp. 265–267, Wuxi, China, October 2018.

[35] X. Xu, H. Xu, X. Zhu, Y. Li, L. Jia, and S. Li, "Collision-free path planning of unmanned aerial robots based on a algorithm," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pp. 5635–5640, Shenyang, China, June 2014.

[36] Y. Zeng, D. You, X. Ai, X. Zhang, S. Wang, and Z. Yang, "Optimal path planning of mobile robot based on improved ant colony algorithm," in *Proceedings of the 2021 4th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, pp. 366–370, Shanghai, China, November 2021.