EAI Endorsed Transactions
on AI and Robotics
Research Article EAI.EU

A Highly Efficient ACO-SA Algorithm for Robot Path Planning

Wei Li [1 2 3], Feng Yang [2], Zhibin Li [3, 4, 5]*, Juan Mou [1], Yanmei Ha [2], Yi Liu[1], Yanmei Qin[1], Peiyang Wei [3, 4]*, Linlin Chen [7], Xun Deng [5], Tinghui Chen [3, 4], Jia Liu [3, 4], Jianhong Gan [3, 4], ZhenZhen Hu [10], Yonghong Deng[8], Guodong Li[9], Qifeng Su[6]

[1] Key Laboratory of Numerical Simulation of Sichuan Provincial Universities, School of Mathematics and Big Data, Neijiang Normal Univeristy, Neijiang 641000, Sichuan Province, China;
[2] School of Urban Rail Transit, Sichuan Railway Vocational College, Chengdu 610097, China;
[3] Dazhou Key Laboratory of Government Data Security, Sichuan University of Arts and Science, Dazhou, Sichuan 635000, China;
[4] School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China;
[5] Xinjiang Technical Institute of Physics &Chemistry, Chinese Academy of Sciences, Urumqi 830011, China
[6] School of Big Data and Artiffcial Intelligence, Chengdu Technological University, Chengdu 611730, China
[7] Renshou No.1 High School, Renshou 620581, China
[8] School of Science, Xihua University, Chengdu 610039, China
[9] Faculty of Artificial Intelligence Education, Central China Normal University, Wuhan 430079, China
[10] College of Communication Engineering, Chengdu University of Information Technology, Chengdu 610225, China

Abstract

Intelligent algorithms continue to develop, and efficient path planning for mobile robots in complex environments relies heavily on such algorithms. This article proposes an ACO-SA path planning method that combines ant colony and simulated annealing to solve the problems of slow iteration and long computation in classical ant colony algorithms. The training base map is gridded and modeled, and the path is initially calculated and parameterized using traditional ant colony algorithms. The simulated annealing cooling mechanism is introduced to optimize the pheromone strategy, and the robustness of the algorithm is tested using a multi-modal large model random map. Simulation shows that under the map of the training base, the path length of the ACO-SA algorithm remains unchanged, and the convergence speed is improved by 88.9% and 58.3% respectively, while the running time is shortened by 1.5% and 3.5% respectively; In the worst results of the random map, the shortest path is shortened by 36.57% and 35.95% respectively compared to the traditional ant colony algorithm. This algorithm has better optimization effect and path stability, providing a practical solution for intelligent detection robot path planning.

Keywords: Path Planning; Mobile Robot; Ant Colony Algorithm; Simulated Annealing Algorithm; Grid Modeling; ACO-SA

Received on 01 December 2025, accepted on 16 December 2025, published on 29 December 2025

doi:10.4108/airo.11177

*Corresponding author. Email: lizhibin@cuit.edu.cn, weipy@cuit.edu.cn

1. Introduction

The penetration of intelligent algorithms in the field of path planning is increasingly deepening. Due to their self organizing, positive feedback, and robustness characteristics, Ant colony algorithm and its improved forms have become key tools for interactive and secure path planning.

However, traditional ant colony algorithms still have shortcomings in terms of dynamic environment adaptability, convergence efficiency, and local optimal avoidance problems, which also drives academia to improve their comprehensive performance through algorithm fusion or strategy optimization.

In the field of robotic path planning for redundant manipulators, Khan et al. [1] addressed key challenges such as model dependency, computational intensity, and real-time performance limitations in trajectory tracking. They proposed a novel model-free control algorithm, ZNNBAS, which integrates a Zeroing Neural Network (ZNN) with the Beetle Antennae Search (BAS) meta-heuristic approach.

In the field of path planning and hybrid optimization algorithms, scholars have proposed optimization solutions for different scenarios: Yogita et al. [2] proposed a hybrid ACO-PSO technique for robot path planning, which aims to find an optimal collision-free path by synergizing the advantages of ant colony optimization and particle swarm optimization. Li et al. [3] integrated 2-opt with ACO to shorten cultural and tourism routes by 3.1 km, providing a practical reference for robot path optimization through the combination of local search improvement and pheromone-based path selection. Wen et al. [4] developed a hybrid GA-SA algorithm for mobile robot path planning, adopting grid method for environmental modeling. Tanira et al. [5] proposed an improved Dijkstra approach with a simple sampling method for robot navigation, which avoids the exhaustive search of the traditional Dijkstra algorithm to save time and resources. Liu XJ et al. [6] designed an improved multi-objective path planning method for substation inspection robots to solve the problems of insufficient safety, unsmooth paths and unbalanced multi-objective optimization of the traditional ACO algorithm.

Yang et al. [7] proposed a dual population ACO strategy that reduces the number of iterations of the TSP problem by 30%, leveraging parallel computation of two distinct ant colonies to balance exploration and exploitation. Ijaz et al. [8] developed a bio-inspired Beetle Antennae Search (BAS) algorithm for real-time path planning and PID controller tuning of differential mobile robots, achieving simultaneous trajectory tracking and obstacle avoidance through optimized control parameters. Mo et al. [9] constructed a multi-colony ant optimization system based on the pheromone fusion mechanism of cooperative games. Xin et al. [10] proposed a Self-Evolving Particle Swarm Optimization framework for dynamic path planning. By converting particle-wise operations to tensor operations via Tensor Operation Form (TOF) to enhance computational efficiency. Yen et al. [11] proposed a fuzzy ant colony optimization (FACO) algorithm, which minimizes the iterative learning error of ACO using fuzzy control. Xin et al. [12] proposed a reformative bat algorithm (RBA) for mobile robot path planning, the algorithm achieves success rates of 93.33% and 90% in two different environments, with satisfactory real-time performance and path planning effect verified by ROS-based real-world robot navigation. Wei et al. [13] proposed a fusion algorithm of improved ant colony and DWA for dynamic path planning. Using GRRT-Connect to allocate initial pheromones unequally, adding ant colony relay search and slice optimization to obtain the global optimal path, and adopting adaptive speed adjustment and pre-calculated dynamic obstacle avoidance in DWA. Zou et al. [14] integrated the improved A* algorithm and optimized dynamic window method, establishing a topological layer map on the raster map to remove redundant nodes and optimize path smoothness. Zhao et al. [15] proposed a dual-layer optimization A* algorithm combined with dynamic window method. Yang et al. [16] proposed a ROS-based autonomous control algorithm for warehouse logistics AGV path planning, constructing a warehouse SLAM map and improving traditional A* and DWA algorithms.

Wu et al. [17] proposed an improved artificial potential field method for UAV 3D path planning, introducing the relative distance between UAV and target into the repulsive potential field function to solve the unreachable target problem. Wei et al. [18] proposed an improved genetic algorithm for inter-frame correlation smooth path planning, generating candidate paths through random and directional search. Xu et al. [19] formulated UAV 3D path planning as a multi-constraint optimization model (integrating no-fly zones and terrain features) and proposed a solution based on the Crested Porcupine Optimization (CPO) algorithm. Wang ZH et al. [20] proposed an adaptive moth-flame optimization algorithm for 3D path planning, introducing dynamic flight direction adjustment and position intersection strategies to avoid local optima. Wu et al. [21] designed a hierarchical path planning method based on hybrid topology-grid-metric maps, dividing the environment into grid sub-maps to construct a topological framework. Jiao et al. [22] developed a Safe Fast Jump Point Search (SFJPS) algorithm, redefining jump point judgment criteria to avoid oblique obstacle crossing, adopting angle-based search direction priority to reduce redundant nodes.

Li E et al. [23] proposed an improved bidirectional ant colony algorithm for static environment global path planning, using obstacle effective vertices as encounter conditions for bidirectional search. Li XY et al. [24] integrated MOPSO with DWA for path planning in complex dense obstacle environments, establishing a multi-obstacle coverage model to judge dense obstacle areas. Liu B et al. [25] proposed an improved PSO and fitness function for dynamic and unknown environments, adopting circular search for next moving points, sectional collision judgment for new obstacles, and path replanning, with strong predictive ability in path planning. Zhang Y et al. [26] optimized the ant colony algorithm for QR code mobile robots in structured grid environments, limiting ant search directions to horizontal and vertical. Wang F et al. [27] improved PSO for urban logistics UAV path planning, using grid method for environment modeling to establish a multi-constraint model (shortest path, height variation, minimum grid risk).

Ouyang et al. [28] proposed a hierarchical path planning method based on hybrid genetic particle swarm optimization (HGA-PSO), using the triangle method for space environment modeling. Zhang R et al. [29] integrated RRT* and DWA for dynamic obstacle avoidance in complex environments, improving RRT* to generate global optimal safe paths (eliminating dangerous nodes). Han et al. [30] applied adaptive PSO to UAV dynamic task allocation and path planning in a rolling horizon framework, integrating energy management (modeling flight, computation and recharging consumption) and adopting adaptive perturbation strategies and simulated annealing-based local search. Huang et al. [31] proposed a fusion algorithm (IHHO-IDWA) of improved Harris Eagle Algorithm and improved DWA. Wu et al. [32] hybridized simulated annealing for intelligent vehicle path planning, proposing a new node connection mode and switch mechanism to obtain suboptimal paths in shorter time, improving path accuracy compared with four-connection A* algorithm, especially in low obstacle percentage maps.

Yang S et al. [33] optimized the ant colony algorithm for forest fire prevention robots based on extended neighborhoods, extending the search neighborhood from 8 to 10, improving heuristic functions with path length and energy consumption. Kong et al. [34] enhanced 3D UAV path planning with an improved ant colony algorithm, constructing a gravitational potential field to initialize pheromones. Wang J et al. [35] developed an ACO-SA hybrid algorithm for air defense target assignment, establishing a firing advantage degree evaluation model and integrating the merits of ACO and SA, reducing computational complexity and meeting the solving requirements of target assignment in air defense combat. Wang FR et al. [36] designed a distributed multi-robot collaborative path planning method based on improved ant colony algorithm, dividing the problem into path planning and cooperative collision avoidance stages, achieving better coordination among robots. THUKRAL et al. [37] hybridized ACO-SA for TSP solutions, using SA to avoid local minima and improve convergence rate.

Hao et al. [38] enhanced the ant colony algorithm with goal heuristic information, introducing distance gain coefficients and weighted distance heuristic factors to amplify target influence. Liu ZY et al. [39] improved the ant colony algorithm for vehicle routing problems, introducing a saving matrix updating selection probability formula, piecewise function for volatilization factors and 2-opt method for local search. MA J et al. [40] fused ant colony and A* algorithms, introducing A*'s evaluation function to improve ACO's heuristic function and pheromone update mode. Zhou XH et al. [41] proposed an improved ant colony algorithm for robot global path planning, using A* to plan initial paths and increase pheromones, introducing smoothness and dual heuristic functions, and establishing automatic pheromone evaporation factor update strategies, generating higher quality and smoother paths in MATLAB 2D grid simulations. Tao Z et al. [42] refined the simulated annealing algorithm for handling robot path planning, using improved grid method for environment modeling and integrating genetic algorithm ideas to define annealing and lattice coefficients, solving local convergence problems and adapting to different environments. Pu XC et al. [43] combined improved PSO with ant colony algorithm for mobile robot multi-goal path planning, transforming target selection into TSP (optimized by ACO) and path planning between targets into optimization problems.

The above algorithms have been studied for robot path planning problems using different algorithms. This study proposes a global path planning algorithm for intelligent inspection robots based on ant colony algorithm combined with simulated annealing algorithm to address the efficiency and adaptability deficiencies of existing path planning methods in complex environments.

Section 2 introduces the basic principles of the ant colony algorithm and the simulated annealing algorithm. Section 3 presents the implementation principles and methods for robot path planning based on ACO-SA. Section 4 conducts simulations and result analysis of the ACO-SA algorithm. Section 5 summarizes the beneficial effects of the algorithm and outlines future directions for improvement.

2. Ant Colony And Simulated Annealing Algorithm

2.1. Ant Colony Algorithm

Ant colony algorithm is a swarm intelligence optimization algorithm inspired by the foraging behavior of ants in nature. In the process of searching for food, ants will release pheromones along the path, and the higher the concentration of pheromones, the greater the probability of being chosen by other ants [44,47].

Heuristic information η_{ij} is usually taken as the reciprocal of the distance between nodes i and j. In the two-dimensional planar path planning of robots, the most commonly used distance is Euclidean distance, which directly reflects the actual geometric distance between nodes.

$$d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2},\quad \eta_{ij}=\frac{1}{d_{ij}} \tag{1}$$

In a path planning problem, suppose that in a graph with n nodes, the probability of an ant moving from node i to node j is determined by the following formula (p_{ij}^k) [48,49]:

$$p_{ij}^k=\begin{cases}\dfrac{\left[\tau_{ij}\right]^\alpha\left[\eta_{ij}\right]^\beta}{\sum\limits_{s\in allowed_k}\left[\tau_{is}\right]^\alpha\left[\eta_{is}\right]^\beta} & ,j\in allowed_k\\ 0 & ,otherwise\end{cases} \tag{2}$$

The above formula, τ_{ij} represents the concentration of pheromones on the path between node i and node j; η_{ij} is a heuristic factor, usually taken as $\eta_{ij}=\frac{1}{d_{ij}}$, where d_{ij} is the distance between node i and node j; α and β respectively represent the relative importance of pheromone heuristic factors and expected heuristic factors, used to control the relative importance of pheromones and heuristic factors; $allowed_k$ represents the set of nodes that ant k can choose from in the next step.

After completing a traversal, ants update their pheromones according to the following formula [50,51]:

$$t_{ij} = (1-r)t_{ij} + \Delta t_{ij}$$
$$\Delta t_{ij} = \sum_{k=1}^{m} \Delta t_{ij}^{k} \quad (3)$$

The above formula, ρ is the volatilization coefficient of pheromones, with a range of values between [0,1], used to simulate the natural volatilization of pheromones over time; m is the number of ants; $\Delta\tau_{ij}^{k}$ represents the increment of pheromones left by the ant k on path (i, j). If ant k passes through path (i, j), then $\Delta\tau_{ij}^{k}=\frac{Q}{L_k}$, Q constant, where L_k is the length of the path traversed by ant k.

In order to more accurately evaluate the quality of the path, this paper uses the ant cycle model to calculate the pheromone increment $\Delta\tau^{k}{}_{ij}$. In this model, the amount of pheromones Q released by a single ant is inversely proportional to the total length L_k of its constructed path (obtained by accumulating Euclidean distances), thereby guiding the ant colony to converge towards a shorter path.

$$Q_{ij}^{k} = \begin{cases} Q/L, & \text{if edge } (i, j) \text{ is on ant k's current path} \\ 0 \end{cases} \quad (4)$$

The basic procedure of the ant colony algorithm is as follows:

(i)Initialize Parameters: Set parameters including the number of ants, initial pheromone concentration, pheromone evaporateon coefficient, pheromone heuristic factor, expected heuristic factor, etc. Initialize the pheromone matrix.

(ii)Ant Traversal: Each ant selects the next node based on the state transition probability formula, starting from the origin and incrementally constructing a path until reaching the destination.

(iii)Pheromone Update: After all ants complete traversal, update the pheromone concentration on paths according to the pheromone update formula.

(iv)Check Termination Conditions: If the termination condition is met (such as reaching the maximum number of iterations or path convergence), the algorithm ends and outputs the optimal path; Otherwise, return to step 2 to continue iterating.

2.2. Simulated Annealing Algorithm

The simulated annealing algorithm originates from the simulation of the solid annealing process, and its basic idea is to perform random search in the solution space, balancing global and local search capabilities by controlling temperature parameters.

During the algorithm execution, a relatively high initial temperature T_0 is first set, and as the algorithm progresses, the temperature gradually decreases. At each temperature, perturb the current solution to generate a new solution, and calculate the difference ΔE between the objective function of the new solution and the current solution. If ΔE<0, accept the new solution; If $\Delta E \geq 0$, the new solution is accepted with probability ：

$$P=e^{-\frac{\Delta E}{T}} \quad (5)$$

where T is the current temperature. This probability acceptance mechanism enables the algorithm to accept poor solutions with a high probability in the early stages of the search, jumping out of local optima. As the temperature decreases, the probability of accepting poor solutions gradually decreases, and the algorithm gradually converges to the global optimum [52,55].

The main steps of simulated annealing algorithm are as follows:

(i)Initialization: Determine the initial solution x_0, initial temperature T_0, cooling rate, and termination temperature.

(ii)Search at current temperature: At the current temperature T, perturb the current solution x to generate a new solution x_{new}, calculate the difference in the objective function ΔE, and decide whether to accept the new solution based on the acceptance criteria.

(iii)Cooling: Reduce the temperature according to the cooling rate α_T, that is：

$$T = T \, . \, \alpha_T \quad (6)$$

(iv)Determine the termination condition:If the current temperature $T \leqslant T_{end}$, the algorithm ends and outputs the optimal solution; Otherwise, return to step 2 to continue searching.

3. Robot Path Planning Based on ACO-SA

3.1. ACO-SA Algorithm Fusion Strategy

This paper integrates the SA mechanism into the ACO algorithm. After each iteration of the ACO algorithm, the SA algorithm is used to optimize the current best path obtained [56].

The specific fusion strategy is as follows:

(i)Post-ACO Iteration: After each ACO iteration is completed, the current best path is obtained.

(ii)SA Initialization: This best path is used as the initial solution for the SA algorithm. Suitable initial temperature, cooling rate, and termination temperature are set.

(iii)SA Optimization: The SA algorithm perturbs this initial solution to generate a new path. Based on the SA acceptance criterion, it decides whether to accept the new path.

(iv)Feedback to ACO: After optimization by the SA algorithm, an improved path is obtained and fed back to the ACO algorithm. It improved path is used to update the pheromone concentration.

(v)Dynamic Adjustment of ρ: The pheromone evaporation coefficient (ρ_k) in the ACO algorithm is dynamically adjusted based on the SA temperature changes:

$$\rho_k = 1 - e^{-\frac{T}{T_0}} \tag{7}$$

(vi)High Temperature : ρ is appropriately increased to accelerate pheromone updating, enabling the algorithm to explore new paths more rapidly.

(vii)Cooling Down : ρ is gradually decreased to enhance algorithm convergence and prevent premature convergence to a local optimum.

3.2. Implementation of ACO-SA in Robot Path Planning

Selecting the college training base as the specific research object, the map of the training base is then processed in a gridded manner, dividing it into multiple equally sized grid cells, each representing an independent state. These grid cells are further divided into feasible and infeasible grids, where feasible grids represent the passable areas within the training base and infeasible grids represent the areas where obstacles are located within the training base.

Figure 1. Training Base

The grid corresponding to the obstacle is black, represented by "1" in the matrix; The grid corresponding to the passable blank area is white, represented by "0" in the matrix [57].

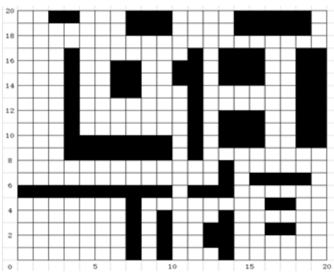


Figure 2. 20x20 Grid Map

The robot path planning based on ACO-SA mainly includes the following steps:

Algorithm ACO-SA	
Input:	$G, K, M, S, E, \alpha, \beta, \rho, Q, T_0, T_{min}, \alpha_T, \boldsymbol{\tau}$
Operation	
	/* Initialization */
Initialize:	$G, K, M, S, E, \alpha, \beta, \rho, Q, T_0, T_{min}, \alpha_T, \boldsymbol{\tau}$
Set:	α=1, β=30, Q=1, T_0=100, T_{min}=1e^{-8}, α_T=0.95
	M=100, K=50
	/* Main Loop */
for	m=1 **to** M **do**
for	k=1 **to** K **do**
Part 1:	Initialization of individual ant state
Part 2:	Filter the optional next nodes of the current node
Part 3:	Multiple ants' path searching
	Calculate p_{ij}^k with (2)
Part 4:	Update ant status
	Update τ_{ij} , $\Delta\tau_{ij}$ with (3)
Part 5:	Record current path → SA improvement: avoid local optimum
if	PL < PL_{min} **do**
	PL_{min} = PL;
end	
if	PL > PL_{min}& $e^{-\frac{\Delta E}{T}}$ > rand ( ) **do**
	PL_{min} = PL;
end	
else	
	PL = 0;
end if	
end for	
Part 6:	Pheromone update → SA improvement: dynamically updating pheromones [58,59]
	$\rho_k = 1 - e^{-\frac{T}{T_0}}$;
	$\boldsymbol{\tau} = (1 - \rho_k).\boldsymbol{\tau} + \Delta\tau_{ij}$;
Part 7:	Annealing termination → SA condition limitation
	$T = T.\alpha_T$;
if	$T < T_{min}$ **do**
	$T = T_{min}$;
end	
end for	
	/* Output Results */
Output:	Best path and its length

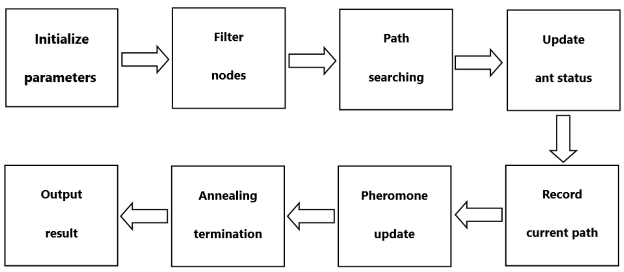


Figure 3. Algorithm flowchart.

4. Algorithm Simulation Verification and Analysis

4.1. A. Simulation Environment Settings

To verify the effectiveness of the ACO-SA algorithm, simulation experiments are conducted on the MATLAB 2024a platform. The hardware environment used for simulation is Intel Core i5 processor, 256GB of memory, and the software environment is Windows 10 system. Build a simulated map of the training base, with a size of 20x20, containing obstacles of different shapes and distributions. Set the initial values of the parameters for the ACO-SA algorithm: ant count m=50, iteration count K=100, pheromone importance parameter α=1, heuristic factor importance parameter β=1, pheromone volatilization coefficient ρ=0.5, initial temperature T_0 =100 for the simulated annealing algorithm, temperature decay coefficient T_α=0.95, and termination temperature $T_{min}=e^{-8}$. At the same time, the parameters of the traditional ant colony algorithm are set to be partially the same as those of the ant colony algorithm in the ACO-SA algorithm for comparison [60,62].

Table 1 Algorithm key Parameter Settings.

Algorithm	**α**	**β**	**ρ**	**Remarks**
M1	1	1	0.5	ACO1
M2	1	9	0.7	ACO2
M3	1	30	0.7	ACO3
M4	1	9	0.4～1	VACO1
M5	1	30	0.4～1	VACO2
M6	1	9	0～1	ACO-SA1
M7	1	30	0～1	ACO-SA2

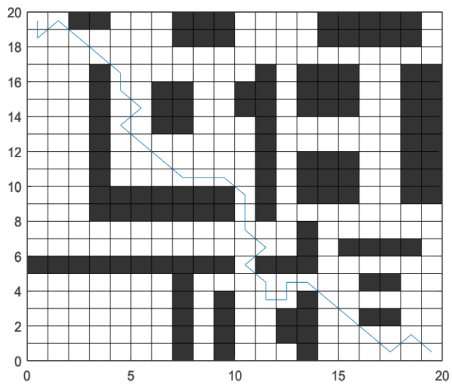
Table notes: ACO - Traditional Ant Colony Algorithm, VACO - Improved Ant Colony Algorithm, ACO-SA - Ant Colony Annealing Improved Algorithm.

Run ACO-SA algorithm, traditional ant colony algorithm, and improved ant colony algorithm multiple times, and record the path length and convergence iteration times obtained each time. Perform statistical analysis on multiple simulation results to obtain the average path length and average convergence iteration time.

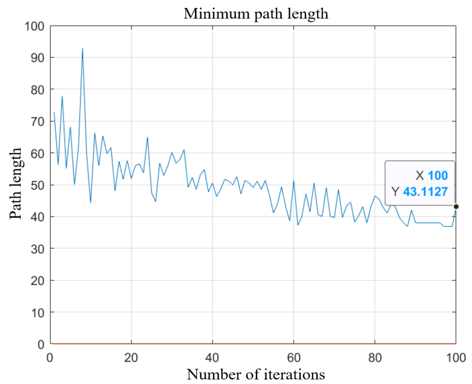
In the analysis of the results of the first random simulation test, there are significant differences in the performance of each algorithm. Among them, M3 and M7 algorithms

4.2. Simulation Result Analysis

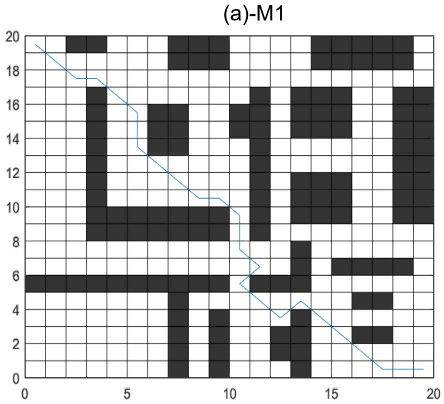
performed the most outstandingly, not only successfully finding the optimal path among all algorithms (path length of 32.6274), but also converging very quickly, reaching a stable state in only 4 iterations, demonstrating efficient optimization ability and stability; In contrast, the M1 algorithm performs at the bottom, failing to converge to a feasible solution within the set upper limit of 100 iterations, exposing its insufficient adaptability or low search efficiency in complex random environments.



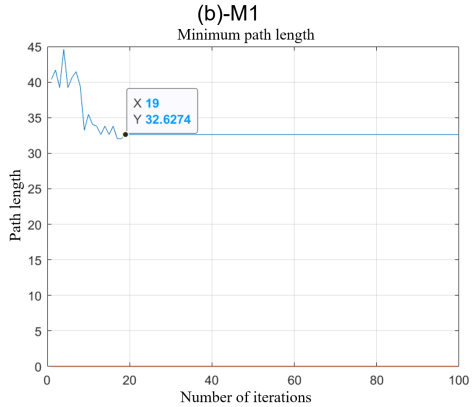
(a)-M1



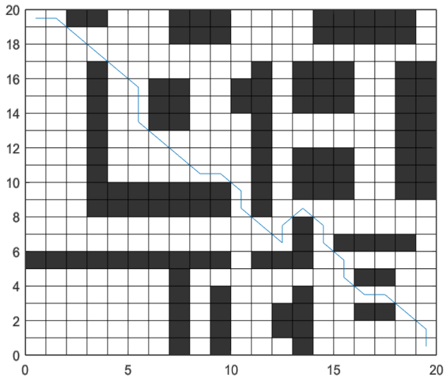
(b)-M1



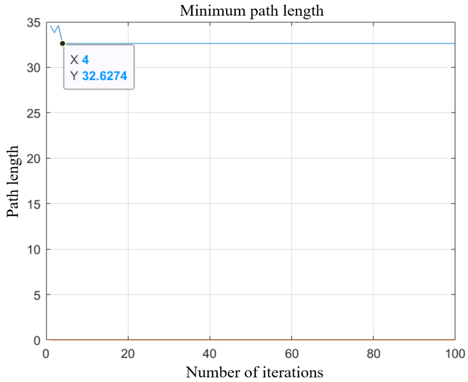
(c)-M2



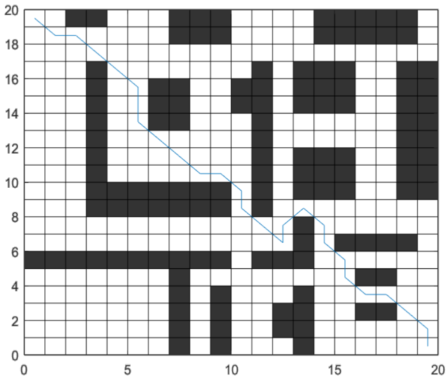
(d)-M2



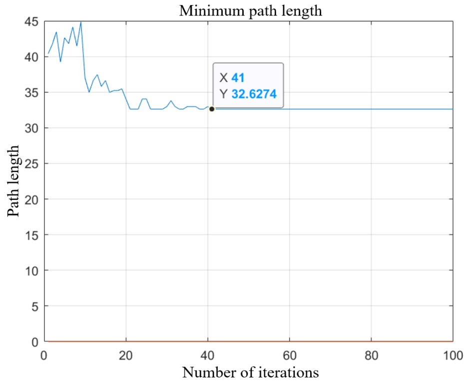
(e)-M3



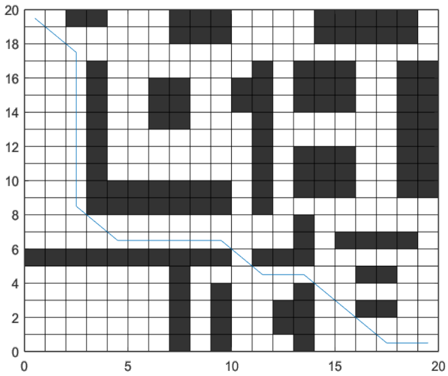
(f)-M3



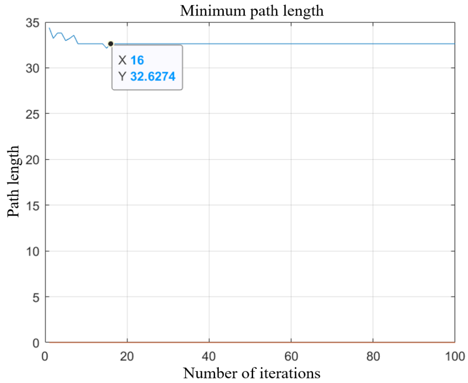
(g)-M4



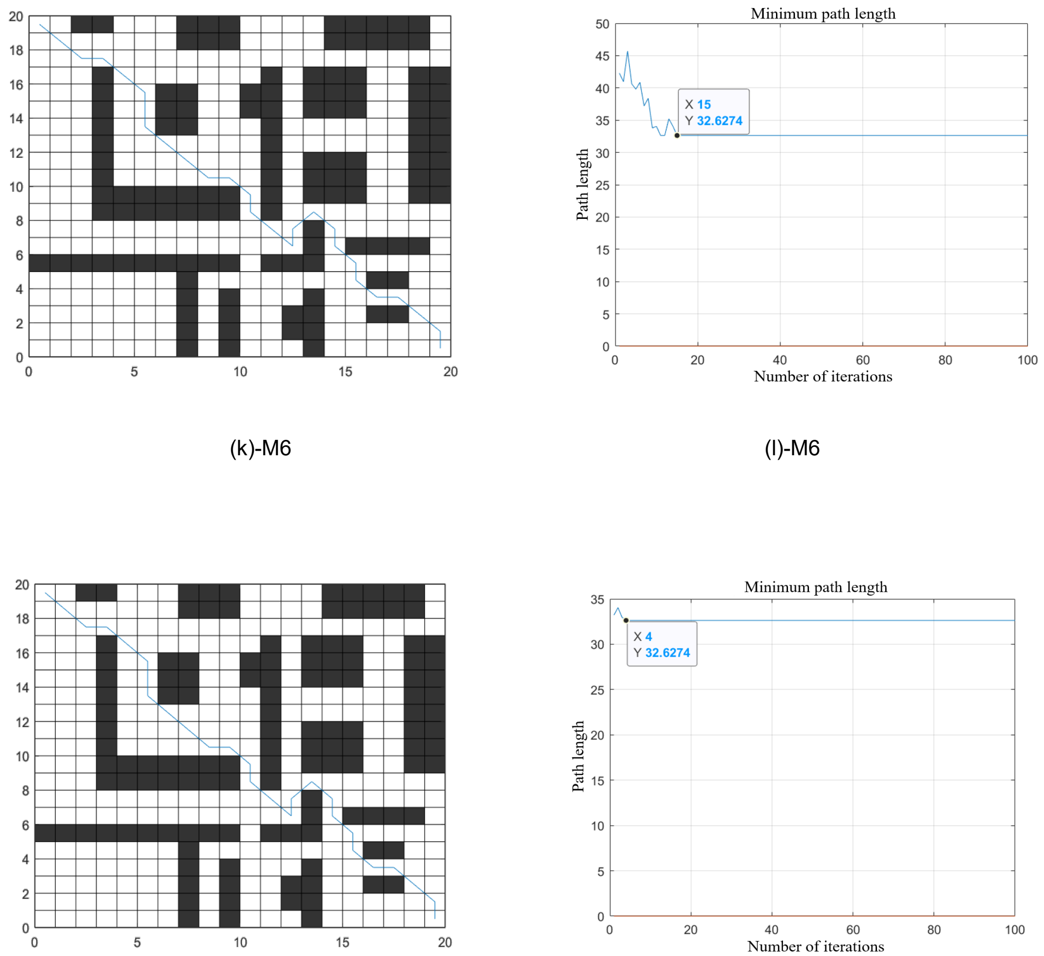
(h)-M4



(i)-M5



(j)-M5



(k)-M6

(l)-M6

(m)-M7

(n)-M7

Figure 4. The paths found and the convergence algebras calculated by M1-M7(for the first time)

The simulation results show (as shown in Figures 5 and 6) that under the same conditions (α=1, β=30), the average path length planned by the ACO-SA algorithm is basically the same as that of the ACO algorithm and VACO algorithm. As shown in Table 2, the average path length obtained by the traditional ant colony algorithm is 32.86, while the average path length obtained by the ACO-SA and VACO algorithms is 32.63. In terms of convergence speed, compared to ACO and VACO algorithms, the convergence speed of ACO-SA algorithm is improved by 88.9% and 58.3% respectively (as shown in Table 2: the average convergence algebra obtained by ACO algorithm is 45, the average convergence algebra obtained by VACO algorithm is 12, and the average convergence iteration number obtained by ACO-SA algorithm is 5). It indicates that the ACO-SA algorithm can find the optimal path faster and significantly improve the convergence speed.

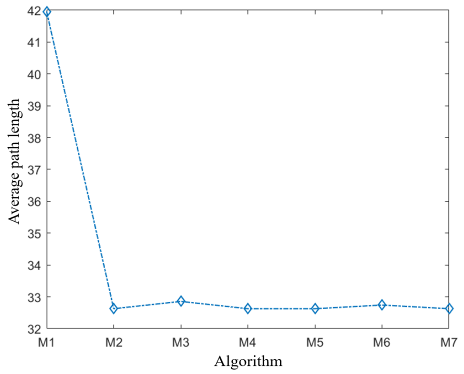


Figure 5. Line graph of algorithm average path length

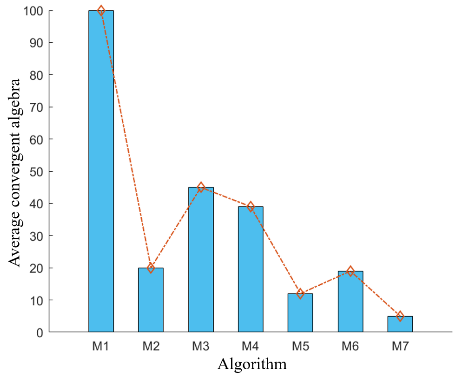


Figure 6. Algebraic bar chart of algorithm average convergence

During the path planning process, one observes the convergence process of each algorithm separately. The traditional ant colony algorithm cannot achieve convergence within 100 generations, while the improved ant colony algorithm exhibits large fluctuations in iteration results, a slow convergence speed, and a tendency to get stuck in local optima, resulting in the final path not being globally optimal. [63]. Due to the introduction of simulated annealing mechanism, The ACO-SA algorithm can quickly explore the solution space in the early stage of iteration, avoiding premature falling into local optima. As the iteration progresses, the temperature of the simulated annealing algorithm gradually decreases, and the algorithm gradually converges to the global optimal solution, resulting in a shorter path length, a more stable convergence process, and a faster convergence speed.

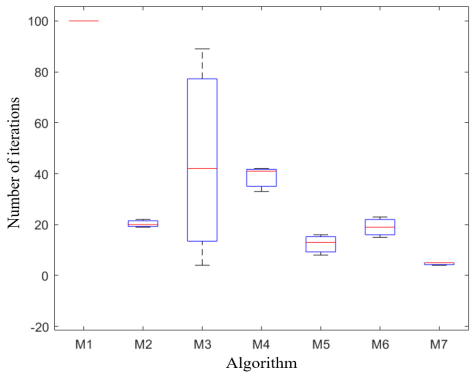


Figure 7. Algebraic box plot of algorithm convergence

In addition, by adjusting the parameters of the ACO-SA algorithm, the path length and iteration times are calculated for different values of alpha. It is found that the algorithm performed better when the alpha value is between 30-40, as shown in Figure 8-10.

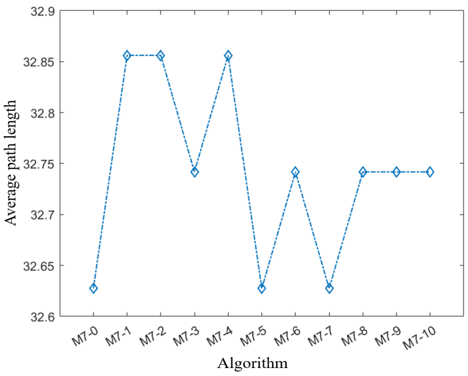


Figure 8. Line graph of algorithm average path length

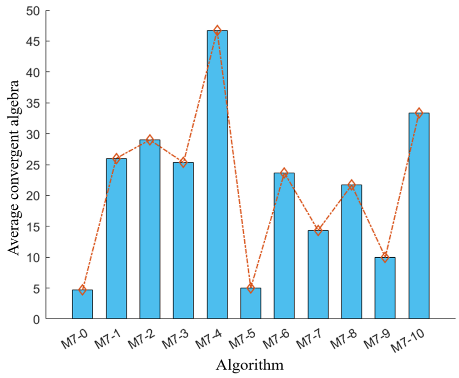


Figure 9. Algebraic bar chart of algorithm average convergence

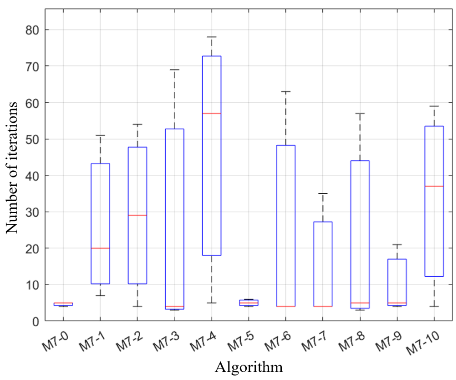


Figure 10. Algebraic box plot of algorithm convergence

In summary, the ACO-SA algorithm has significant advantages over traditional ant colony algorithms in terms of path length and convergence speed. Especially when α=30, the algorithm has stronger iterative convergence ability and lower path randomness (compared to α=35), which can provide better path planning solutions.

Table 2 Comparison of algorithm simulation results

Algorithm	L(1)	A(time)	R	T(s)	remarks
M1	41.96	100	high	20.80	non convergence
M2	32.63	20	high	17.39	convergence
M3	32.86	45	in	16.27	convergence
M4	32.63	39	high	16.56	convergence
M5	32.63	12	in	16.62	convergence
M6	32.74	19	high	16.37	convergence
M7	32.63	5	low	16.03	convergence

Table 3 Comparison of algorithm simulation results

Algorithm	L(1)	A(time)	R	T(s)	remarks
M7-0	32.63	5	low	16.03	α=30
M7-1	32.86	26	in	15.96	α=31
M7-2	32.86	29	in	15.64	α=32
M7-3	32.74	25	in	15.72	α=33
M7-4	32.86	47	in	17.47	α=34

M7-5	32.63	5	in	14.75	α=35
M7-6	32.74	24	in	17.01	α=36
M7-7	32.63	14	in	15.38	α=37
M7-8	32.74	22	low	16.72	α=38
M7-9	32.74	10	in	16.68	α=39
M7-10	32.74	33	in	16.51	α=40

Table notes: L-average path length, A-average convergence algebra, R-path randomness, T-running time.

In addition, two maps are randomly generated using an AI multimodal large model to simulate the algorithm's search for the optimal path in dynamic maps.

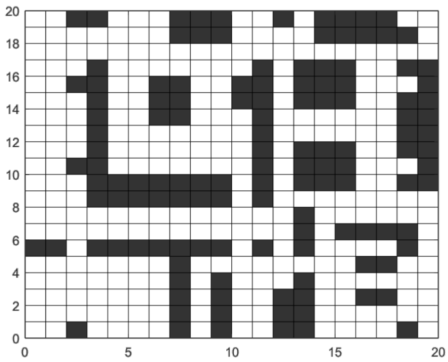


Figure 11. The first random map

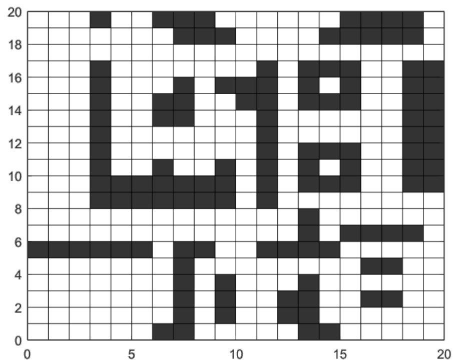
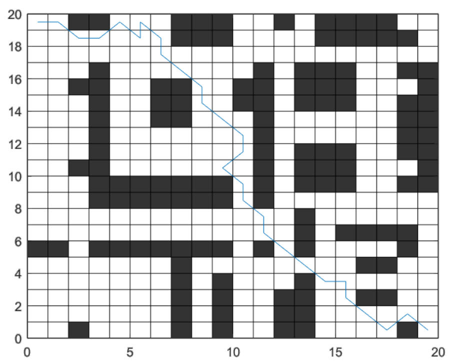
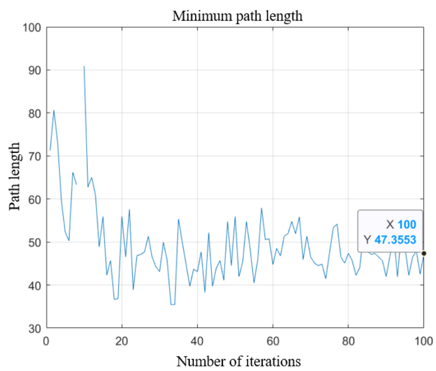


Figure 12. The second random map

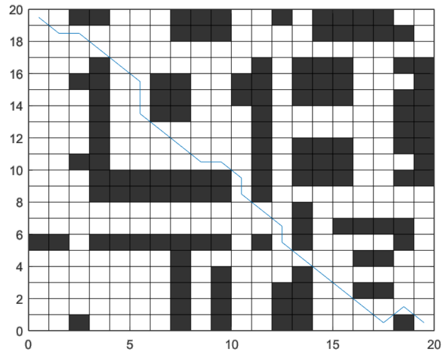
In this experiment, a random map 1 generated by a multimodal large model is used as the testing environment to simulate and evaluate the performance of algorithms M1 to M7. After comprehensive analysis of the results of various algorithms, the simulation result with the worst performance is selected. This result indicates that the M1 algorithm performed the worst in this test, as it failed to converge and effectively find feasible solutions or achieve optimization goals throughout the 100 generation evolution process; In contrast, although M5 algorithm ultimately found the same optimization path as M7, its convergence process took significantly longer and required 12 iterations to reach a stable state, while M7 algorithm only needed 7 iterations to complete convergence, demonstrating higher optimization efficiency and faster convergence speed.



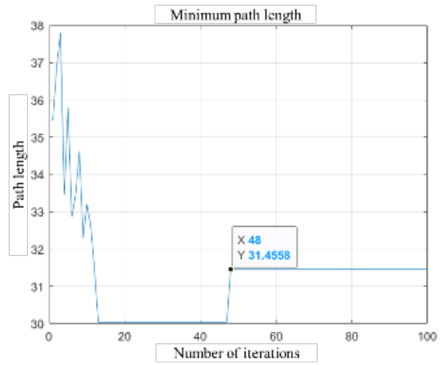
(a)-M1-Map 1



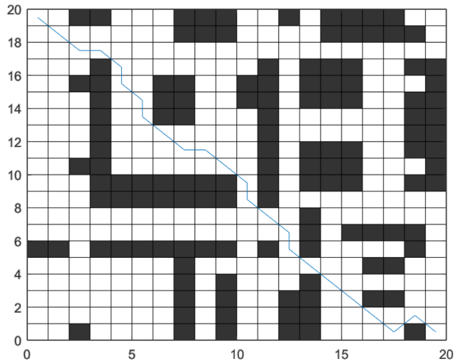
(b)-M1-Map 1



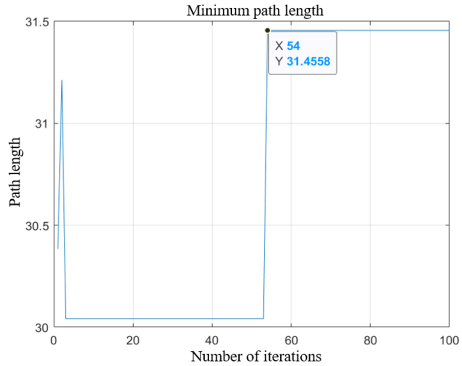
(c)-M2-Map 1



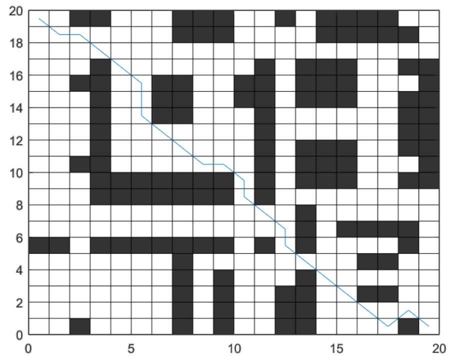
(d)-M2-Map 1



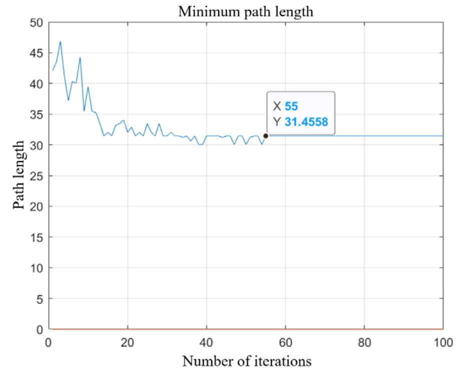
(e)-M3-Map 1



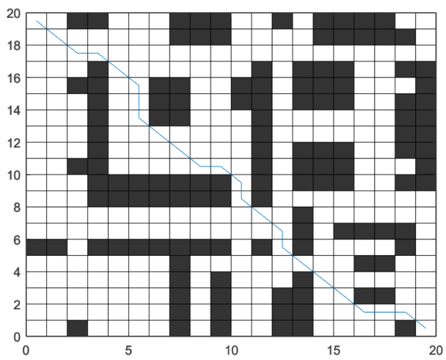
(f)-M3-Map 1



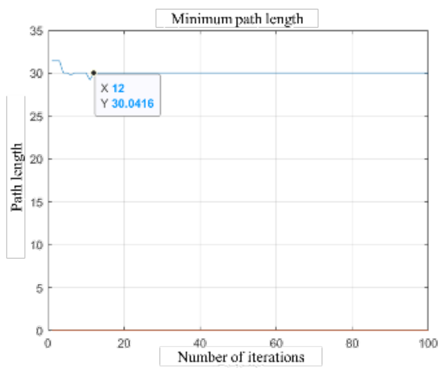
(g)-M4-Map 1



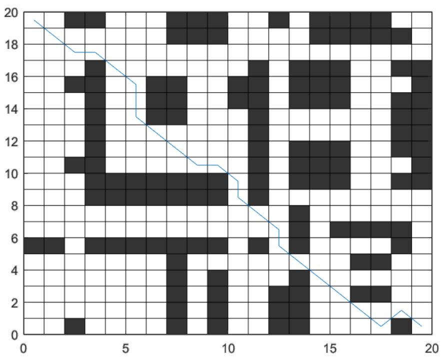
(h)-M4-Map 1

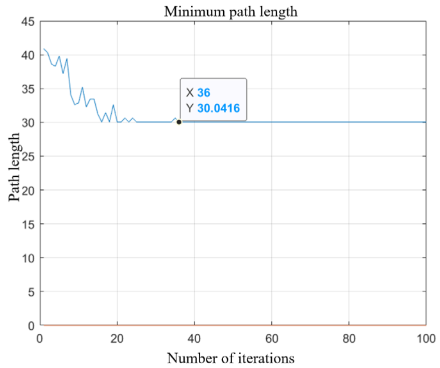


(i)-M5-Map 1



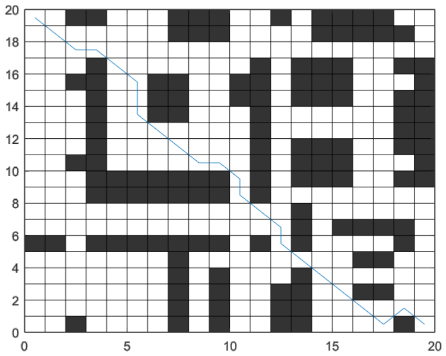
(j)-M5-Map 1

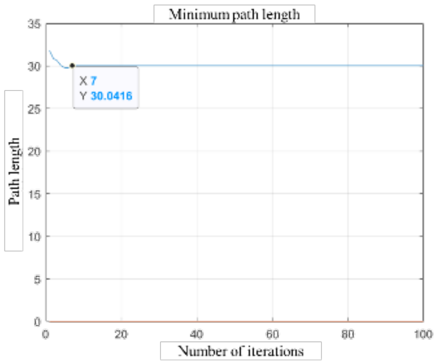




(k)-M6-Map 1

(l)-M6-Map 1



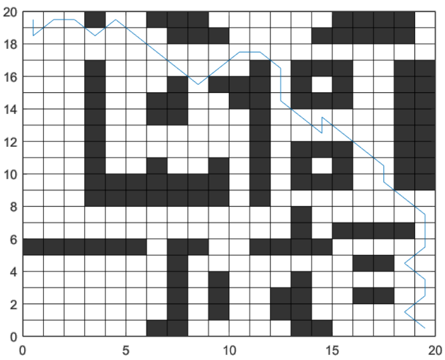


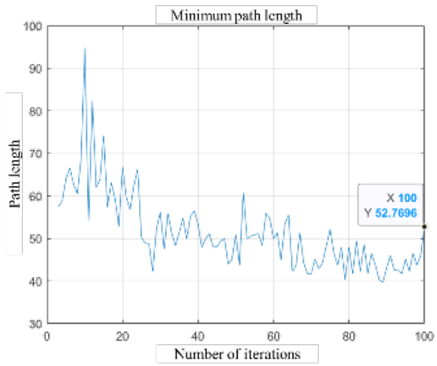
(m)-M7-Map 1

(n)-M7-Map 1

Figure 13. The map1 paths found and the convergence algebras calculated by M1-M7(for the first time)

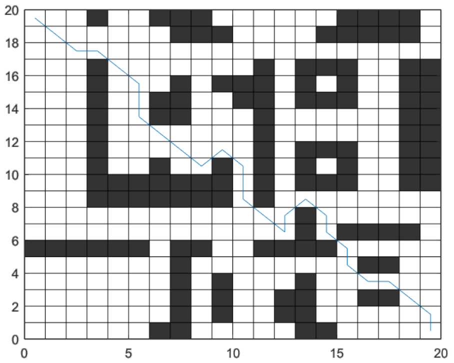
Similarly, using Random Map 2 as the testing environment to evaluate the performance of algorithms M1 to M7. The simulation results with the worst performance are selected. The M1 algorithm performed the worst, as it failed to converge and effectively find feasible solutions or achieve optimization goals throughout the 100 generations of evolution; Although the M5 algorithm ultimately found the same optimization path as M7, it required 22 iterations to reach a stable state, while the M7 algorithm only needed 16 iterations to converge.

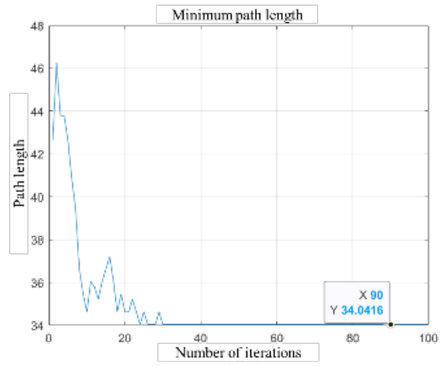




(**a**)-M1-Map 2

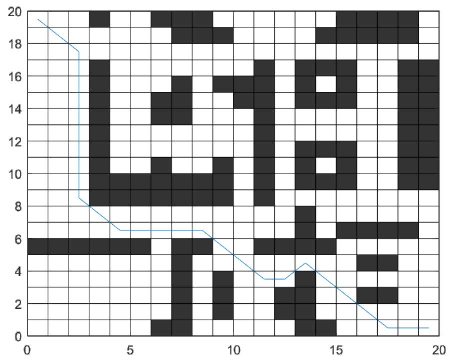
(**b**)-M1-Map 2

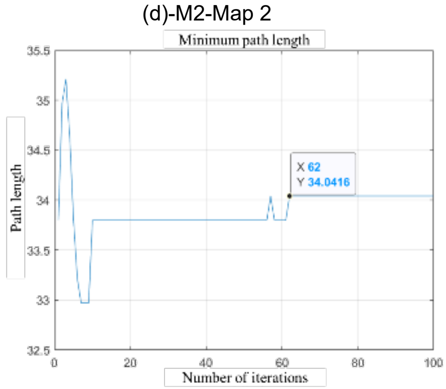




(c)-M2-Map 2

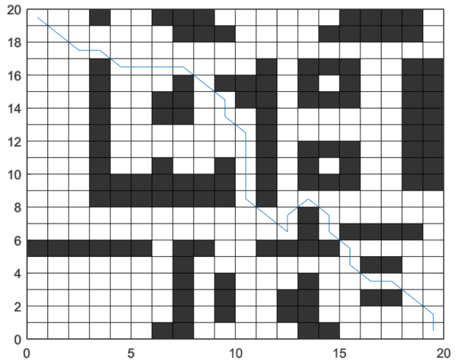
(d)-M2-Map 2

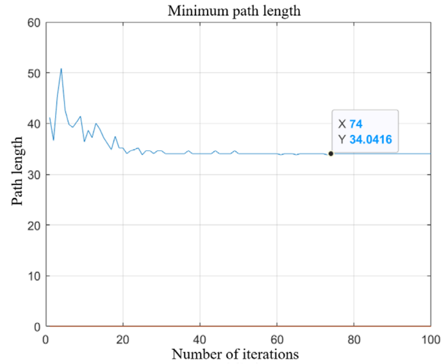




(e)-M3-Map 2

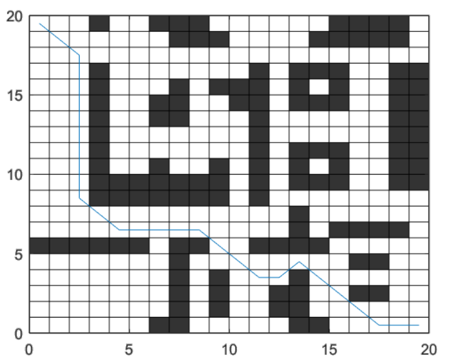
(f)-M3-Map 2

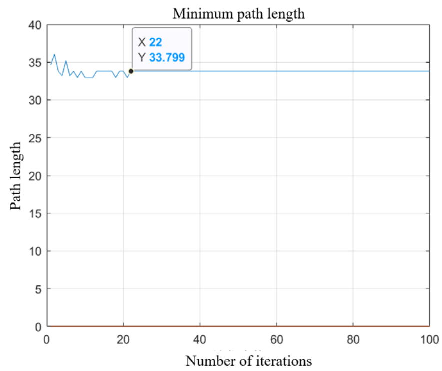




(g)-M4-Map 2

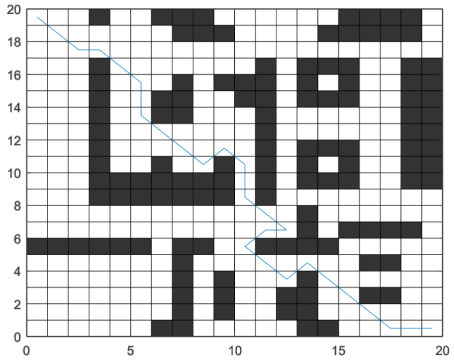
(h)-M4-Map 2



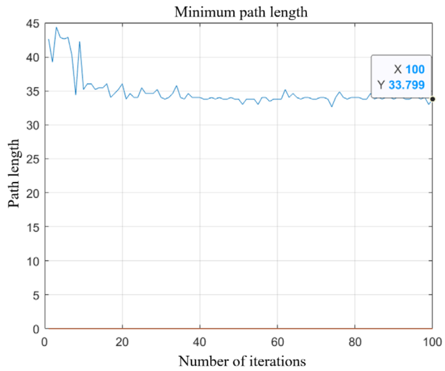


(i)-M5-Map 2

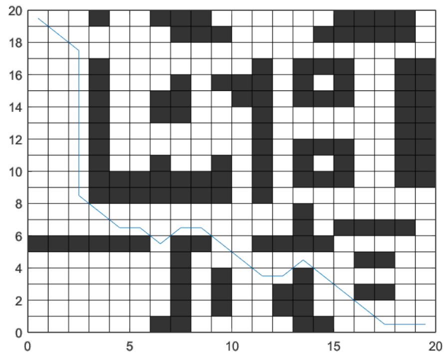
(j)-M5-Map 2



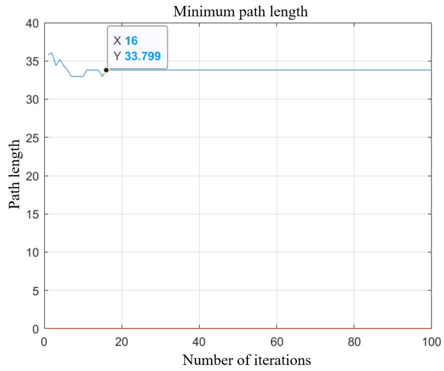
(k)-M6-Map 2



(l)-M6-Map 2



(m)-M7-Map 2



(n)-M7-Map 2

Figure 14. The map2 paths found and the convergence algebras calculated by M1-M7(for the first time)

The obtained results through simulation experiments clearly indicate that compared with the classic traditional ant colony algorithm (M1), the ACO-SA algorithm (M7) also demonstrates significant performance advantages in terms of path length and convergence speed; In the comparison of the worst results, the shortest path of the ACO-SA algorithm in the map 1 scenario is 36.57% shorter than that of the traditional ant colony algorithm, and only required 7 convergence epochs to complete convergence. However, the shortest path of the traditional ant colony algorithm is longer and did not converge after 100 convergence epochs. In the map 2 scenario, the shortest path of the ACO-SA algorithm is also shortened by 35.95% compared to the traditional ant colony algorithm, achieving convergence in only 16 convergence epochs. The traditional ant colony algorithm also had a longer path and did not converge after 100 convergence epochs; This algorithm not only performs significantly better than traditional ant colony algorithms at the worst performance level, but also, after multiple repeated statistics and averaging the results obtained, the ACO-SA algorithm solved for different maps. In the map 1 scenario, its shortest path is shortened by 30.38% compared to traditional ant colony algorithms, with only 5 convergence iterations. In the map 2 scenario, its shortest path is shortened by 28.84% compared to traditional ant colony algorithms, with only 12 convergence iterations. However, traditional ant colony algorithms still have longer paths and fail to converge after 100 convergence iterations in these two map scenarios, fully reflecting the characteristics of ACO-SA algorithm always presenting the shortest path length and the least algorithm time consumption.

Table 4 Comparison of Worst Results in Random Map Algorithm Simulation

Algorithm		M1	M2	M3	M4	M5	M6	M7
Map1	L(1)	47.36	31.46	31.46	31.46	30.04	30.04	30.04
	A (time)	100	48	54	55	12	36	7

Map2	L(1)	52.77	34.04	34.04	34.04	33.80	33.80	33.80
	A (time)	100	90	62	74	22	100	16

Table 5 Comparison of simulation results of random map algorithms

Algorithm		M1	M2	M3	M4	M5	M6	M7
Map1	L(1)	43.15	30.51	30.51	30.51	30.04	30.04	30.04
	A (time)	100	28	20	35	8	28	5
Map2	L(1)	47.50	33.88	33.88	33.96	33.80	33.80	33.80
	A (time)	100	47	27	63	20	95	12

5. Conclusions and prospective research

This article proposes a path planning method for intelligent inspection robots based on ant colony algorithm. Integrating ant colony and simulated annealing algorithm effectively overcomes the drawbacks of traditional ant colony algorithm, such as slow iteration and susceptibility to local optima. By using grid modeling to process the training base map and utilizing a multimodal large model to generate diverse maps, the proposed ant colony annealing algorithm is examined. Introducing simulated annealing mechanism to optimize ant colony path and dynamically adjust pheromone volatilization coefficient [64]. Simulation results show that the ACO-SA algorithm has shorter paths and faster convergence, providing an effective solution for training global path planning for intelligent inspection robots.

However, path planning problems in real-world scenarios are more complex, involving not only dynamic obstacles but also factors such as multi robot collaboration and task priority [65]. Future research plans to integrate attention enhanced dynamic decision-making, multimodal environment modeling, and adaptive strategy optimization algorithms used in large models such as reinforcement learning and DeepSeek in dynamic environments, further enhancing the path optimization capability of ACO-SA algorithm to better adapt to complex and changing practical application scenarios. In addition, we will explore the integration of large-scale distributed collaborative frameworks and multi-agent attention mechanisms to extend the algorithm to the field of multi robot collaborative path planning. By leveraging the global situational awareness and dynamic task allocation capabilities of large-scale models, we aim to enhance the overall detection efficiency and intelligence level [66,67].

Funding

This research is funded by the Key Laboratory of Numerical Simulation of Sichuan Provincial Universities (Grant 2025SZFZ005, 2025SZFZ009) and the Open Project of Dazhou Key Laboratory of Government Data Security (Grant ZSAQ202515), and also supported by Sichuan Railway Vocational College, and also supported by the National Funded Postdoctoral Research Program GZC20241900, Natural Science Foundation Program of Xinjiang Uygur Autonomous Region 2024D01A141, Tianchi Talents Program of Xinjiang Uygur Autonomous Region and Postdoctoral Fund of Xinjiang Uygur Autonomous Region, the open project of Dazhou Key Laboratory of Government Data Security under grants ZSAQ202501, ZSAQ202502, ZSAQ202505 and ZSAQ202507.

References

[1] Khan AT, Cao X, Li Z, Li S. Evolutionary Computation Based Real-time Robot Arm Path-planning Using Beetle Antennae Search. EAI Endorsed Transactions on AI and Robotics.2022;1.

[2] Yogita G, Kavita C, et al. A Hybrid ACO-PSO Technique for Path Planning. 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE. 2015.

[3] Li SM, Li YR, et al. Design of Jiaxing Cultural Tourism Routes Based on Improved Ant Colony Algorithm. Scientific and Technological Innovation.2025;8:64-67.

[4] Wen JY, Huai BW, et al. Research on Path Planning for Mobile Robot Based on Grid and Hybrid of GA/SA. Advanced Materials Research.2012; 479-481:1499-1503.

[5] Tanira AH, Abuhadrous IMI. An Improved Sampling Dijkstra Approach for Robot Navigation and Path Planning. International Journal of Intelligent Systems and Applications.2023;15:51-64.

[6] Liu XJ, Gao HB, et al. Multi Objective Path Planning Method for Substation Inspection Robots Based on

Improved Ant Colony Algorithm Automation and Instrumentation. 2025;40 (11):43-49,79.
[7] Yang C, Li GC, et al. Optimized Ant Colony Algorithm for Solving TSP with Dual Populations under Pheromone Fusion Mechanism. Journal of Xiamen University of Technology. 2025;33 (03):26-36.
[8] Ijaz MU, Khan AT, & Li S. Bio-inspired BAS: Run-time Path-planning and The Control of Differential Mobile Robot. EAI Endorsed Transactions on AI and Robotics,1.
[9] Mo Y, You X, Liu S. Multi-Colony Ant Optimization Based on Pheromone Fusion Mechanism of Cooperative Game. Arabian Journal for Science and Engineering.2021;47: 1657-1674.
[10] Xin JH, Li Z, et al. Efficient Real-Time Path Planning with Self-Evolving Particle Swarm Optimization in Dynamic Scenarios. Unmanned Systems. 2024;12(02):215-226.
[11] CT Yen, MF Cheng. A Study of Fuzzy Control with Ant Colony Algorithm Used in Mobile Robot for Shortest Path planning and obstacle avoidance. Microsystem Technologies.2018;24:125-135.
[12] Xin G, Shi L, et al. Mobile Robot Path Planning with Reformative Bat Algorithm. Plos One.2022;17(11): e0276577.
[13] Wei LX, Zhang YK, et al. Robot Dynamic Path Planning Based on Improved Ant Colony and DWA Algorithm. Control and Decision Making.2022; 37(9): 2211-2216.
[14] Zou W, Han BC. Path Planning with The Improved A* Algorithm and Optimized Dynamic Window Method is Integrated. Computer Integrated Manufacturing Systems.2024;30(1):184-195.
[15] Zhao W, Wu ZY. Dynamic Path Planning of Double-layer Optimization A* Algorithm and Dynamic Window Method. Journal of Computer Engineering & Applications.2021; 57(22):295.
[16] Yang GH, Wei JL. Path Planning of Logistics Robot Based on Improved A* and DWA Algorithms. Science Technology & Engineering.2022; 22(34):15213.
[17] Wu Q, Zeng Q, et al. Research on The Application of Improved Artificial Potential Field Method in UAV Path planning. Journal of Chongqing University of Technology. 2022; 36(3):144-151.
[18] Wei T, Long C. Path Planning of Mobile Robots Based on Improved Genetic Algorithms. Journal of Beihang University.2020;46(4):703-711.
[19] Xu GH, Deng Y, et al. Improved 3D Path Planning of UAV with Optimized Algorithm of Crested Porcupine. Journal of South-Central Minzu University. 2022;44(2):237.
[20] Wang ZH, Dai YQ. Three-Dimensional Path Planning Based on Adaptive Moth Fighting Optimization Algorithm. Application Research of Computers. 2023;40(1):63.
[21] Wu X, Yang JJ, et al. Hierarchical Path Planning of Mobile Robots for Composite Maps. China Mechanical Engineering.2023;34(5):563.
[22] Jiao SM, Liang JY, et al. An Improved Path Planning Algorithm for Jump Point Search Mobile Robot. Information & Control. 2025;54(3), 525-535.
[23] Li E, Qi QQ. Path Planning of Mobile Robots with Improved Two-Way Ant Colony Algorithm. Journal of Computer Engineering & Applications.2021;57(18), 281.
[24] Li X Y, Qu Y, Zhang Y. DWA Path Planning Algorithm Based on Multi-target Particle Swarm in Complex Environment. Journal of National University of Defense Technology. 2022;44(4):52.
[25] Liu B, Chen T, et al. Path Planning Based on Improved Particle Swarm Optimization in Dynamic and Unknown Environments. Application Research of Computers. 2014;31(6):1673.
[26] Zhang Y, Li K, et al. Path Planning Method of QR Code Mobile Robot Based on Improved Ant Colony Algorithm. Journal of Chongqing University of Posts and Telecommunications. 2021;33(3):491.
[27] Wang F, Yang QP. Route Planning of Urban Logistics Unmanned Aerial Vehicle Based on Improved Particle Swarm Optimization Algorithm. Science Technology and Engineering.2023;23(30): 13187-13194.
[28] Ouyang HB, Quan YB, et al. Hierarchical Path Programming Method Based on Hybrid Genetic Particle Swarm Optimization Algorithm. Journal of Zhengzhou University. 2020;41(4):34.
[29] Zhang R, Zhou L, et al. Dynamic Path Planning for Mobile Robots Integrating RRT * and DWA Algorithms. Journal of System Simulation. 2024;36 (4): 957-968.
[30] Han Z, Guo WA. Dynamic UAV Task Allocation and Path Planning with Energy Management Using Adaptive PSO in Rolling Horizon Framework. Electronic Communications, IOT and Big Data. 2025;15(8):4220.
[31] Huang Z F, Liu Y H, et al. Research on Mobile Robot Dynamic Path Planning Based on Improved Harris Hawk Algorithm and Improved Dynamic Window Algorithm. Journal of Computer Applications. 2024;41(2):450-458.
[32] Wu LL, Yang JH, et al. Global Path Planning of Intelligent Vehicles Based on Hybrid SA Algorithm. Journal of Jiangsu University. 2019;40(3):249.
[33] Yang S, Hong T, et al. Path Planning of Forest Fire Prevention Mobile Robot with Improved Ant Colony Algorithm. Forest Engineering. 2024;40(1):152.
[34] Kong LW, Wang F, et al. Three-Dimensional Path Planning of UAVs Based on Improved Ant Colony Algorithm. Electro-Optics and Control.2023;30 (3): 63.
[35] Wang J, GAO XG, et al. Solving Algorithm for TA Optimization Model Based on ACO-SA. Journal of Systems Engineering and Electronics.2011;22(4): 628-639.
[36] Wang FR, Du L, et al. Distributed Multi-robot Collaborative Path Planning Based on Improved Ant Colony Algorithm. Journal of South-Central Minzu University.2023; 42(5): 650.
[37] Thukral S, Diwaker C. ACO-SA: Enhanced Optimization for Tsp. International Journal of Advanced Research in Computer Science, 2017;8(7): 294.
[38] Hao ZM, An PJ, et al. Mobile Robot Path Planning Based on Enhanced Goal Heuristic Information Ant Colony Algorithm. Science Technology and Engineering. 2023;23(22): 9585-9591.
[39] Liu Z Y, Zhao LX, et al. Research on Improved Ant Colony Algorithm for Vehicle Path Problem. Journal of Hebei University of Science & Technology. 2022;43(1): 80.
[40] MA J, Song SJ, et al. Path Planning of Mobile Robots Using Ant Colony-A* Algorithm. Journal of Xi'an Polytechnic University. 2020;34(1): 72.
[41] Zhou XH, Li YQ, et al. Robot Path Planning Based on Dual Heuristic Information Ant Colony Algorithm. Journal of System Simulation.2025;37(5): 1280-1289.
[42] Tao Z, Lei ZB, et al. Path Planning of Handling Robot Based on Improved Simulated Annealing Algorithm. Computer Measurement & Control. 2018;26(7): 182.
[43] Pu XC, Li JJ. Mobile Robot Multi-Goal Path Planning Using Improved Particle Swarm Optimization. Journal of Intelligent Systems. 2017; 12 (3): 301-309.
[44] Zhu JY, Gao MT. AUV Path Planning Algorithm Fusing Particle Swarm and Improved Ant Colony Algorithm.

Journal of Computer Engineering & Applications.2021;57(6): 267.
[45] Zhang TR, Wu BK, et al. Research on Improved Ant Colony Algorithm for Global Path Planning of Robots. Journal of Computer Engineering & Applications.2022;58(1):282.
[46] Duan HB, Wang DB. Improved Ant Colony Algorithm and Simulation for Fast Global Optimization. Information and Control.2004;33(2): 241-244.
[47] Zhou JD, Gao WZ, et al. Path Planning of Mobile Robot Based on Improved Ant Colony Algorithm. Science Technology & Engineering. 2022;22(28): 12484.
[48] Shi SC, Li R, et al. Assembly Sequence Planning Based on Improved Ant Colony Algorithm. Computer Integrated Manufacturing Systems.2010;16(6): 0.
[49] Qin GL, Yang JB. An Improved Ant Colony Algorithm Based on Adaptively Adjusting Pheromone. Information And Control.2002;31(3): 198-201,210.
[50] Zhang C, Li Q, et al. Improved Ant Colony Optimization Based on Particle Swarm Optimization and Application. Chinese Journal of Engineering. 2013;35(7): 955-960.
[51] Zhu HD, Sun Z, et al. Path Planning of Mobile Robot Based on Improved Ant Colony Algorithm. Journal of Chongqing University of Posts and Telecommunications. 2016;28(6): 849.
[52] Zhu HD, Zhong Y. An Improved Simulated Annealing Algorithm. Computer Technology and Development. 2009; 19(6): 32-35.
[53] Zhang LB, Yao ZX, et al. Fast Simulated Annealing Algorithm and Application. Oil Geophysical Prospecting.1997; 32(5): 654-660.
[54] Jiang LC, Liu J. Simulated Annealing Algorithm and improvement. Chinese Journal of Engineering Geophysics. 2007; 4(2): 135-140.
[55] Yao X, Cheng GL. Simulated Annealing Algorithm and Application. Journal of Computer Research and Development.1990;27(7): 1-6.
[56] Liu B, Meng PS. Solving Traveling Salesman Problem Using Simulated Annealing-Based Ant Colony Algorithm. Journal of Huazhong University of Science and Technology.2009; (11): 26-30.
[57] Yu C, Qiu QW. Hierarchical Robot Path Planning Algorithm Based on Grid Map. Journal of University of Chinese Academy of Sciences. 2013;30(4): 528.
[58] Zhang LY, Wang Y, et al. Low-Carbon Logistics Path Optimization Using Chaotic Perturbation Simulated Annealing Ant Colony Algorithm. Computer Engineering and Applications.2017;53(1): 63-68.
[59] Du ZX, Wang ZQ, et al. Improved Polymorphic Ant Colony Algorithm Based on Quadratic Annealing Mechanism. Journal of Central South University.2011;42(10): 3112-3117.
[60] Liu YX, Wang P, et al. Reverse Ant Colony Algorithm Based on Simulated Annealing Strategy. Microcomputer Information.2006;(12S): 265-267.
[61] Ma SD, Gong GH, et al. Ant Colony-Simulated Annealing Algorithm for Target Assignment and Improvement. Journal of Systems Engineering and Electronics.2011;33(5): 1182-1186.
[62] Lin HX, Xiang D, et al. A Review of Path Planning Algorithms for Mobile Robots. Journal of Computer Engineering & Applications.2021;57(18): 38.
[63] Zhang YL, Niu XM. Simulation Study of Ant Colony Algorithm in Mobile Robot Path Planning. Computer Simulation.2011;28(6): 231-234.
[64] Yuan JQ, Li S, et al. Robot Path Planning Method Based on Simulated Annealing Ant Colony Algorithm. Computer Simulation. 2019;36(10): 329-333.
[65] Zhang ZY, Gong SY, et al. Research on Multi-Robot Task Assignment and Path Planning Algorithm. Journal of Harbin Engineering University. 2019;40(10): 1753-1759.
[66] Qu, Y, Li, Z, et al. Robust Robotic Arm Calibration via Multi-Distance Optimization Approach and Lagrange Starfish Algorithm. EAI Endorsed Transactions on AI and Robotics.2025; 4.
[67] Vayadande K, Singh V K, et al. MSCSO: A Hybrid Nature-Inspired Algorithm for High-Dimensional Traffic Optimization in Urban Environments. EAI Endorsed Transactions on AI and Robotics.2025; 4.