

# Multi-strategy adaptable ant colony optimization algorithm and its application in robot path planning

Junguo Cui <sup>a,b</sup>, Lei Wu <sup>a,b,c,\*</sup>, Xiaodong Huang <sup>a,b</sup>, Dengpan Xu <sup>a,b</sup>, Chao Liu <sup>a,b</sup>, Wensheng Xiao <sup>b</sup>

<sup>a</sup> College of Mechanical and Electrical Engineering, China University of Petroleum (East China), Qingdao 266580, China

<sup>b</sup> National Engineering Research Center of Marine Geophysical Prospecting and Exploration and Development Equipment, China University of Petroleum (East China), Qingdao 266580, China

<sup>c</sup> School of Civil and Environmental Engineering, Maritime Institute @NTU, Nanyang Technological University, Singapore 639798, Singapore



## ARTICLE INFO

### Keywords:

Path planning  
Ant colony optimization algorithm  
Directional mechanism  
Adaptive updating

## ABSTRACT

As a widely used path planning algorithm, the ant colony optimization algorithm (ACO) has evolved into a well-developed method within the realm of optimization algorithms and has been extensively applied across various fields. In this study, a multi-strategy adaptable ant colony optimization (MsAACO) is proposed to alleviate the insufficient and inefficient convergence of ACO, employing four-design improvements. First, a direction-guidance mechanism is proposed to improve the performance of node selection. Second, an adaptive heuristic function is introduced to decrease the length and number of turns of the optimal path solutions. Moreover, the deterministic state transition probability rule was employed to promote the convergence speed of ACO. Finally, nonuniform pheromone initialization was utilized to enhance the ability of ACO to select advantageous regions. Subsequently, the major parameters of the strategies were optimized and their effectiveness was validated. MsAACO was proposed by combining these four strategies with ACO. To verify the advantages of MsAACO, five representative environment models were employed, and comprehensive experiments were conducted by comparing them with existing approaches, including the A\* algorithm, variants of ACO, Dijkstra's algorithm, jump point search algorithm, best-first search, breadth-first search, trace algorithm, and other excellent algorithms. The experimental statistical results demonstrate that MsAACO can efficiently generate smoother optimal path-planning solutions with lower length and turn times and improve the convergence efficiency and stability of ACO compared to other algorithms. The generated results of MsAACO verified its superiority in solving the path-planning problem of mobile robots.

## 1. Introduction

The purpose of robot path planning is to obtain an optimal path solution for a robot from the starting node to the target node in a given engineering environment. Robot path planning has wide applications in several fields, including modern robotic systems [1], unmanned aerial vehicles (UAVs) [2], automated guided vehicle systems (AGVs) [3], additive manufacturing (AM) [4], recycling networks of electric vehicles (EVs) [5], and platform pipeline planning [6]. In general, path planning requires establishing an environmental model to simulate the actual engineering environment, relying on path search algorithms to find feasible paths under constrained conditions, and obtaining the optimal path through fitness functions.

Compared to using the experience of engineering designers, utilizing

path-planning algorithms for robot path planning is more efficient and has attracted numerous researchers to study optimization algorithms of path planning, such as bio-inspired neural network algorithm (BINN) [7], particle swarm optimization (PSO) [8], simulated annealing (SA) [9], grey wolf optimization (GWO) [10], A\* algorithm [11], ant colony optimization (ACO) [12], jump point search algorithm (JPS) [13], breadth-first search algorithm [14], and trace algorithm [15].

In addition, numerous researchers have proposed different path-planning algorithms to address the problems of mobile robot path planning. Huang et al. integrated a novel technique with an enhanced neural network algorithm to devise a comprehensive algorithm for handling the dynamic task distribution and route mapping of autonomous underwater vehicles (AUV) [16]. Yu et al. proposed a reinforcement learning-based multi-strategy cuckoo search algorithm to optimize

\* Corresponding author.

E-mail address: [wuleiupc@163.com](mailto:wuleiupc@163.com) (L. Wu).

the path-planning problem of UAVs, which enhanced the convergence efficiency of the algorithm [17]. Wu et al. introduced a deep reinforcement learning technique and proposed autonomous navigation and obstacle avoidance (ANO) to solve obstacle avoidance problems in unmanned surface vehicles [18]. In addition, PSO has a wide application in robot path planning in terms of few parameters and strong optimization performance. To improve the flight accuracy, Zhang et al. proposed an improved PSO called ACVDEPSO by embedding adaptive parameters, different evolution operators, and cylinder vectors into the PSO to obtain an effective path for UAVs in complicated environments [19]. Ahmed et al. proposed an effective trajectory planning method by improving PSO to tackle the full-coverage optimal path-planning problem [20].

Simulated annealing is a global optimization method that utilizes probability techniques to approach a given function. Xiao et al. used enhanced simulated annealing to calculate the region near the optimal path and find unmanned aerial vehicles near the optimal path [21]. Deng et al. combined the SA algorithm and the beetle antennae search (BAS) algorithm to promote the convergence accuracy of the algorithm and heighten real-time performance, thereby achieving the real-time path planning needs of UAVs [22]. However, SA still has the shortcomings of insufficient convergence and difficulty in escaping from local optima. Compared with other heuristic algorithms, although the GWO algorithm has advantages in solving several engineering problems, there are certain shortcomings when facing complex engineering problems. Researchers have proposed improved GWO algorithms to enhance the performance. To improve the capability of the GWO algorithm in handling engineering problems, Yu et al. created a hybrid algorithm called HGWODE, which combines GWO and differential evolution (DE) specifically used for planning UAV flight paths [23]. Wu et al. embedded adaptive convergence factors and adaptive weights into the GWO algorithm and proposed an improved adaptive GWO algorithm (AGWO), that can be effectively used for three-dimensional route planning problems of UAVs in intricate conditions [24]. The Dijkstra's algorithm is a popular algorithm in graph theory and is used to search for the shortest paths between nodes in an environment graph, which can effectively handle path-planning problems [25]. Wu et al. utilized Dijkstra's algorithm to achieve path optimization in urban transportation planning [26]. Compared to the Dijkstra's algorithm, the A\* algorithm has a significant performance with heuristic functions, which can predict the minimum cost between the current node and the target, so that the A\* algorithm can find a more efficient path. To address the path optimization problem of unmanned surface vehicles (USVs), Sang et al. proposed a deterministic algorithm based on an improved multi-objective artificial potential field (MTAPF) by improving the A\* algorithm [27]. Duchon et al. and Akshay et al. solved the robot path optimization problem by improving the search efficiency of the A\* algorithm [28,29]. However, the A\* algorithm needs to store all traversed paths, resulting in low computational efficiency and the requirement of a large amount of memory for large graphs or search spaces. To enhance the computational efficiency of robot path planning, Xiong et al. proposed a specific sample-based path-planning algorithm and improved its convergence accuracy [30]. Zhang et al. integrated the A\* algorithm with a neural network using a unique approach that considers a target-bias search strategy and a new metric function [31]. Kim et al. addressed the problem of path planning by blending a rapid selection process with heuristic information [32].

Genetic algorithm is a global search technique that utilizes natural selection, crossover, and mutation mechanisms to perform path optimization [33]. Xu et al. proposed a missile path-planning genetic algorithm that integrates a twin delayed depth deterministic strategy gradient (GA-TD3) to solve the flight path-planning problem of missiles [34]. Zhou et al. introduced an optimized genetic algorithm into the navigation module for the design of unmanned surface vehicles and their guidance systems to improve convergence speed and recovery efficiency [35]. To improve the efficiency of mobile robots in industrial

production, Jin et al. created a whale optimization algorithm (WOA)-AGA path-planning model using a GA and WOA, which improved the search efficiency of individuals in complex conditions [36]. Chong et al. introduced an improved adaptive genetic algorithm called IAGA to handle the path-planning problems of AUVs [37].

The ACO is a probabilistic algorithm used to search for the optimal path inspired by the foraging behavior of ants in nature [38]. ACO generally utilizes distributed computing and heuristic search systems with characteristics such as global search, parallel processing, robustness, and dynamic adaptation [39]. It has been adapted for various combinatorial optimization issues, including path planning, job network routing, vehicle routing, and image coloring, since it was first employed to address the traveling salesman problem (TSP). Nonetheless, conventional ACO may have limitations, including an insufficient search process, slow convergence rate, and a tendency to become stuck in local optima. To mitigate these shortcomings, researchers enhanced the original ACO algorithm and proposed several variants of ACO algorithm. Yang et al. proposed an improved ACO for rescue path planning called PA-C-IACO [40]. An improved heuristic mechanism, ACO (IHMACO), was proposed by Liu et al. to enhance search efficiency and global search capability by introducing adaptive pheromone concentration and direction judgment heuristic mechanisms [41]. Li et al. proposed an optimized ACO algorithm for solving dynamic and static obstacle path-planning problems [42]. In addition, in response to the premature stagnation issues, Fatemidokht et al. suggested the use of an improved max-min ant system (MMAS) to maintain pheromone concentrations within a specific range [43]. To escape the local optima, Wang et al. promoted the comprehensive search capability of ACO [44]. Miao et al. embedded heuristic information to improve the pheromone and state transition rules in the ACO algorithm, which improved its convergence speed and capability [45]. To improve the convergence speed, accuracy, and efficiency of ACO in solving three-dimensional path optimization, Li et al. introduced an elite potential field and proposed an improved ACO [46]. Tao et al. combined global path planning with fuzzy control and enhanced the convergence speed of ACO [47]. By introducing a 3D path-planning method, Wei et al. alleviated the shortcomings of the original ACO, such as its high computational complexity [48]. Hui et al. proposed a revised ACO to embed a wolf swarm assignment into the ACO to handle route-planning problems [49]. Xiong et al. proposed a novel ACO, called TTACO, in a dynamic environment based on a time taboo [50]. Pei et al. introduced a generalized pheromone update rule and an adaptive parameter adjustment strategy that effectively enhanced the convergence efficiency and accuracy of ACO in path planning [51].

However, the improvements proposed by the above-mentioned studies may have the deficiencies of an insufficient search process and difficulty in escaping from local optima, and their effectiveness will be reduced when it comes to certain complex problems. Thus, shortcomings still exist in improving the capabilities of ACO when dealing with path-planning problems. In this study, an enhanced ACO is proposed to alleviate the shortcomings of the ACO algorithm, and four distinctive mechanisms are presented: direction guidance mechanism, adaptive heuristic function, deterministic state transition probability rule, and nonuniform pheromone initialization. By integrating the four proposed strategies with the original ACO, an innovative and improved ACO called multi-strategy adaptable ant colony optimization algorithm is proposed. A comprehensive parameter optimization for MsAACO was performed to identify the optimal parameter configuration. To establish the effectiveness of MsAACO, comparative analyses with other algorithms were conducted using various instances. The empirical findings highlight the superior performance of MsAACO, particularly in terms of convergence rate and its ability to locate a global optimum solution.

The remainder of the paper is organized as follows: Section 2 describes the environment model and reviews ACO. The four proposed novel strategies are presented in Section 3. Section 4 provides the effectiveness verification of the four proposed strategies. The

comparison experiments and results for verifying the advantages of the proposed MsAACO are also included in Section 4. Section 5 summarizes the main findings of this study and introduces directions for future research.

## 2. Environmental model and the review of ACO

### 2.1. Grid environment model

A mathematical grid model is a simulation environment of a real application space that can be utilized to address the path-planning problems of a robot. Presently, the grid method has wide application in the simulation environment of path planning, which can decrease the complexity of the environmental model. [52,53]. Therefore, the grid method was adopted in these experiments to establish the environmental model. White grids were used to denote free grids, according to the rules of the grid method. The obstacle nodes are represented by black grids. As shown in Fig. 1, the simulated environment application space of the robot was replaced with a grid environment model ( $10 \times 10$ ) in which each grid was arranged with a specific serial number. The target node is denoted as  $T$  and the starting node is represented by  $S$ .

Using the grid method, an environmental model for path planning was established, and all grids were numbered according to the correspondence between the coordinate value and number of grids. The corresponding relationship between the coordinate value and serial number of the grids is expressed as shown in Eq. (1).

$$\begin{cases} x_n = \text{mod}(n, E_x) - 0.5, \\ y_n = R_y + 0.5 - \text{ceil}(n/E_y), \end{cases} \quad (1)$$

where the serial number of the grid is represented as  $n$ ,  $(x_n, y_n)$  are the coordinates of the  $n$ -th grid,  $E_x$  is the number of rows in the environment model,  $E_y$  is the number of columns in the environment model,  $\text{ceil}(\cdot)$  and  $\text{mod}(\cdot)$  are the upward rounding and remainder functions, respectively.

As shown in Fig. 1, for example, when  $n$  is 25,  $E_x$  and  $E_y$  are 10, and the corresponding coordinate values for the grid with serial number 25 are (4.5, 7.5). Therefore, a grid environment model was generated using the grid method to simulate the engineering requirements of the robot movement environment space and handle route-planning problems through feasible paths from the starting node to the target node.

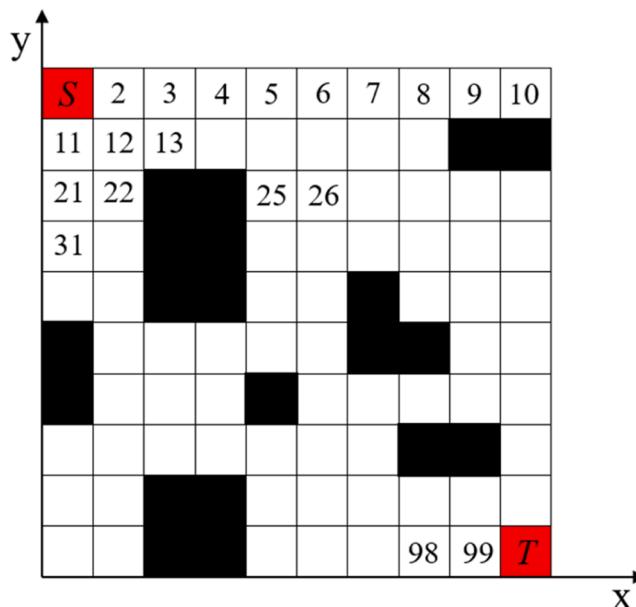


Fig. 1. Grid environment model.

### 2.2. The review of ACO

ACO was proposed based on the foraging behavior of ants and is mainly employed to tackle combinatorial optimization problems, including TSP, vehicle routing problems, and path-planning problems. ACO simulates the mechanism by which ants generate pheromones to determine the shortest path between food sources and nests. The optimal solution can be found by utilizing cooperation between ant individuals and by continuously iterating and updating the pheromones.

Traditional ACO procedures can be divided into two processes: initialization and path searching. Firstly, initialize the search element space, and parameters including iterations  $K$ , the number of individuals  $M$ , pheromone strength  $Q$ , heuristic factor  $\alpha$ , and pheromone evaporation factor  $\rho$ . The roulette wheel is an important approach in the path-searching phase that determines the movement probability of the ant from the  $i$ -th node to the  $j$ -th node for the  $m$ -th ant, and it is expressed as Eq. (2).

$$p_{ij}^m = \begin{cases} \frac{[\tau(i,j)]^\alpha * [\eta(i,j)]^\beta}{\sum_{s \in J_m(i)} [\tau(i,s)]^\alpha * [\eta(i,s)]^\beta}, & (i,j) \in J_m \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where  $\alpha$  and  $\beta$  are the heuristic function and pheromone importance factors, the symbol “\*” represents multiplication, the transition probability of  $i$ -th node to the  $j$ -th node is represented as  $p_{ij}^m$ ,  $J_m$  presents the selectable nodes of ant  $m$  in next movement,  $\tau(i,j)$  presents the pheromone concentration on the node, the visibility of heuristic information is indicated as  $\eta(i,j)$ , which is determined using Eq. (3).

$$\eta(i,j) = \frac{1}{d_{ij}}, \quad (3)$$

where the Euclidean distance of the  $i$ -th node to the  $j$ -th node is represented by  $d_{ij}$ .

After entering the iteration, the pheromone concentration in the feasible path is updated according to its length after an ant individual determines the path. After all individual iterations are completed, the pheromone concentration matrix of the nodes is updated, and the update rules are expressed in Eqs. (4) and (5):

$$\tau_{t+1}^m(i,j) = (1 - \rho) * \tau_t^m(i,j) + \sum_{m=1}^M \Delta \tau_t^m(i,j), \quad (4)$$

$$\Delta \tau_t^m(i,j) = \begin{cases} Q/L_m, & i \text{ to } j \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where  $(1 - \rho)$  is the pheromone residual coefficient with a value between 0 and 1, the pheromone concentration reaching the next point is performed as  $\tau_t^m(i,j)$ ,  $\Delta \tau_t^m(i,j)$  is the concentration variation of the pheromone for the  $m$ -th ant, and the pheromone intensity  $Q$  is a constant. The path length of the  $m$ -th ant is indicated by  $L_m$ .

## 3. The proposed multi-strategy adaptable ACO (MsAACO)

The proposed MsAACO is equipped with four strategies: First, a direction guidance mechanism is proposed to improve the directionality of the node selection. Subsequently, the adaptive heuristic function inserts the location and path information into a heuristic function to decrease the turn times and lengths of the path solutions. Moreover, the deterministic state-transition probability rule provides an innovative node selection method. Finally, nonuniform pheromone initialization enhances the ability of ACO to search for advantageous regions. MsAACO was developed by optimizing the major parameters of the strategies and combining the four strategies with ACO.

### 3.1. Direction guidance mechanism

In the traditional ACO search process, an individual moves toward nearby nodes. As shown in Fig. 2, the current node is represented by D, and there are no obstacle grids around it. The next node can then be selected from the eight surrounding nodes. The next optional nodes are represented by D1–D8. In the position-update rule, the probability of reaching the next node is related to the fitness of the selectable nodes. This implies that the transfer probabilities of the eight nodes must be calculated. According to path-planning theory, the selection of part nodes is redundant and requires unnecessary calculations in the path transition probability of eight nodes, which causes the traditional ACO algorithm to have a long computational time in path planning. Therefore, increasing the direction guidance of node selection and reducing unnecessary choices can alleviate algorithm shortcomings in the position-update rules.

Consequently, a direction guidance mechanism was developed to improve the shortcomings of the original ACO in terms of node selection. Utilizing directional information from the starting node S to the target node T to guide the selection of the next node reduces the original eight optional directions to three and improves the computational efficiency of the algorithms. In Fig. 3(a), the coordinate system is established with the starting node as the origin, and the arrow direction is the direction from the starting node S to the target node T located in the fourth quadrant. The next optional node after the current node was expected to follow the direction of the arrow. Assuming that the co-ordinates of the current node D are  $(x_c, y_c)$  and the co-ordinates of the next optional node  $D_n$  are  $(x_d, y_d)$ , according to the guiding direction mechanism, the value of  $x_{d-c}$  ( $x_{d-c} = x_d - x_c$ ) should not be less than 0 and the value of  $y_{d-c}$  ( $y_{d-c} = y_d - y_c$ ) should not be greater than 0. Therefore, only three nodes (D3, D4, and D5) of the eight optional nodes (D1–D8) shown in Fig. 2 satisfied the requirements. Similarly, when the arrow direction is from the S to the T and points to the second quadrant, the value of  $x_{d-c}$  should not be greater than 0 and  $y_{d-c}$  should not be less than 0. If the arrow direction of node points to the third quadrant, the values of  $x_{d-c}$  and  $y_{d-c}$  should not exceed 0.

In addition, as shown in Fig. 3(b), there are obstacles in the three directions located in the fourth quadrant for the current node D4. Under the guiding direction mechanism, the number of next available nodes was 0. In this case, the node selection method is changed to the roulette-

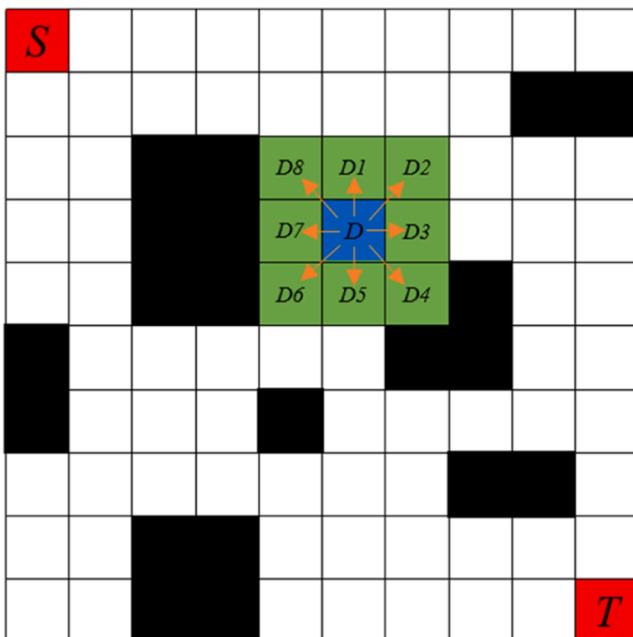


Fig. 2. The selectable nodes of the current node in original ACO.

wheel method. Thus, the next node can be selected from D3, D5, D9, or D10 to prevent the algorithm from falling into a dead cycle. After finding the next node using the roulette-wheel method, the direction guidance mechanism is continuously executed. The direction guidance mechanism reduces the number of optional nodes by adding directional information, thereby reducing computational complexity and improving the efficiency of the algorithm.

### 3.2. Adaptive heuristic function

The ACO capability is influenced by heuristic information. In the original ACO, the heuristic function  $\eta(i,j)$  is defined only as the reciprocal of the distance between the selectable and current nodes, according to Eq. (3). In the partitioned grid environment model, the distance between two randomly adjacent nodes is 1 or  $\sqrt{2}$ . Therefore, the heuristic function of the original ACO lacks guidance, and individual ants randomly select the next node, which causes the ACO to have great blindness in robot path planning. In addition, as the turn times in robot path planning increases, the robot consumes a large amount of energy. Therefore, introducing starting and target node position information, turn times, and path length into the heuristic function can effectively alleviate the above-mentioned drawbacks and improve the purpose and efficiency of the path search.

In this study, an adaptive heuristic function is utilized to enhance the performance of ACO by introducing location information, length, and turns of path solutions. The proposed adaptive heuristic function is expressed by Eq. (6):

$$\eta(i,j)' = \frac{G + H}{d + t} = \frac{G + H}{G * d_{sj} + a * c(i) + H * d_{jt}}, \quad (6)$$

where  $\eta(i,j)'$  indicates the enhanced adaptive heuristic function, the node distance from the start to next is presented as  $d_{sj}$ , and the distance from the next node to the target node is represented as  $d_{jt}$ . The sum of G and H is 1. G represents the weight factor of  $d_{sj}$ , H is the weight factor of  $d_{jt}$ ,  $c(i)$  is the turn time of the last node  $i-1$  to the next node  $j$  in the path, and  $a$  represents the turning factor weight.

The enhanced adaptive heuristic function embeds the distance information and turn times into selectable nodes. The shorter the distance and fewer the turn times of selectable node, the higher the probability of selecting that node. This improves the purposefulness and adaptability of ACO.

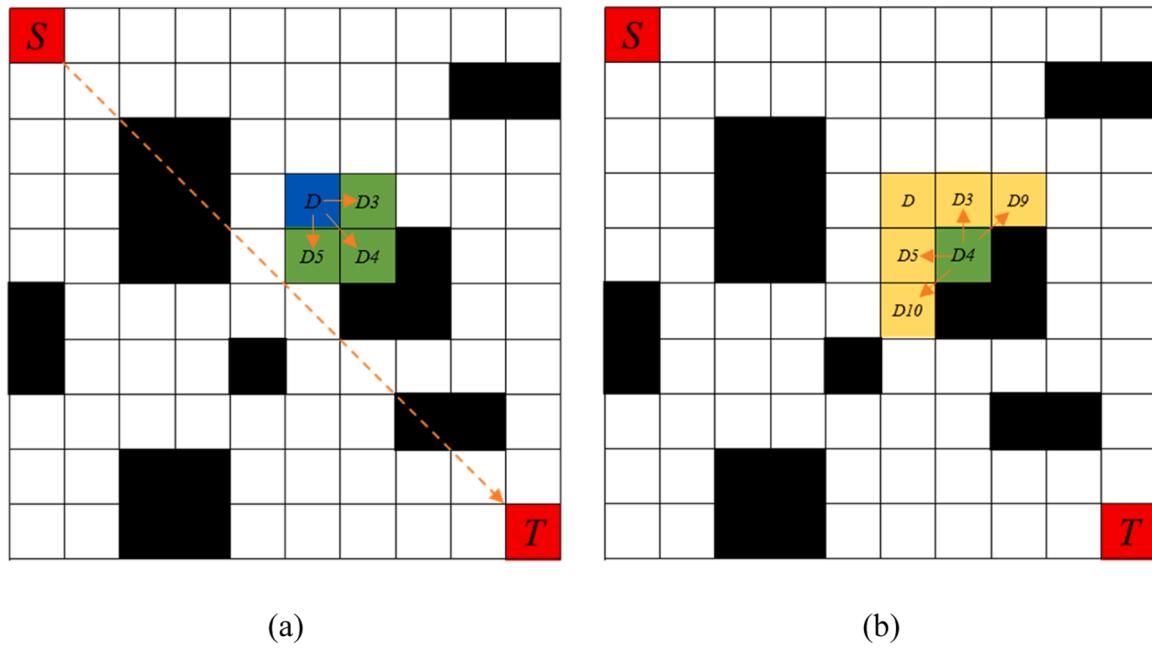
In addition, to enhance the adaptability of the heuristic function, the weight factors of distance  $d_{jt}$  and distance  $d_{sj}$  are introduced into an adaptive adjustment mechanism. Specifically, the value of the weight factor H is positively correlated with the distance  $d_{jt}$ . Because the sum of H and G is 1, after obtaining the value of weight factor H, the value of weight factor G is calculated. The expression of H is shown in Eq. (7).

$$H = -\Delta w * e^{-(-k * d_{jt}) / d_{ST}} + c = -(w_{hmax} - w_{hmin}) * e^{(-k * d_{jt}) / d_{ST}} + w_{hmax}, \quad (7)$$

where  $k$  is a weight coefficient,  $w_{hmax}$  and  $w_{hmin}$  are constants,  $d_{ST}$  is the distance from S to T based on the Euclidean distance, and the distance calculation is expressed as Eq. (8).

$$d_{ST} = \sqrt{(x_T - x_S)^2 + (y_T - y_S)^2} \quad (8)$$

In the original ACO, the value of  $\eta(i,j)$  is referred to as  $1/\sqrt{2}$ . In the adaptive heuristic function, if the distance of the current node from the target node is long, then the value of  $d_{jt}$  is large, and according to Eq. (7), the corresponding value of H is also large. The value of the adaptive heuristic function  $\eta(i,j)'$  decreases according to Eq. (6), thereby improving the convergence speed of the algorithm in the early stages compared with that of the traditional ACO. Similarly, the value of the adaptive function  $\eta(i,j)'$  increases as the values of the weight H



**Fig. 3.** Diagram of the direction guidance mechanism.

and  $d_{iT}$  decrease. Therefore, the accuracy of the algorithm search is improved in the later stages of the iteration. In addition, the adaptive heuristic function improves the adaptability of ACO in robot path planning, which reduces the energy consumption of the robot by reducing the turn times.

### 3.3. Deterministic state transition probability rule

In the original ACO, the roulette wheel is an important way for individual ants to move toward selectable points. In addition, having a large number of optional nodes may increase the computational complexity of traditional ACO. Therefore, a deterministic state transition probability rule is introduced to boost the search efficiency and convergence rate of ACO. Assuming that the position of individual ant  $m$  is at Node  $i$  in  $t$ -th iteration and at Node  $j$  in  $t+1$ -th iteration, the deterministic state probability transition rule from Node  $i$  to Node  $j$  is executed, as shown in Eq. (9).

$$j = \text{argmax} \left\{ [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta \right\}, q \leq q_0 \quad (9)$$

where  $j$  represents the position of the next node to arrive;  $\text{argmax}()$  is a function for determining the position of the maximum value in the matrix;  $q$  is a random number uniformly distributed within the range  $[0, 1]$ . Parameter  $q_0$  is the threshold of  $q$ . From Eq. (9), it can be observed that when the random number  $q$  satisfies the condition, the position of individual  $m$  at the next iteration ( $t+1$ ) is at, where it can generate the maximum product of the pheromone concentration and heuristic information. Therefore, the deterministic selection of nodes, rather than random selection, can help promote the convergence efficiency of the algorithm. In addition, the roulette-wheel method can maintain the global search ability of the original ACO. If  $q > q_0$ , then the deterministic state probability transition rule is replaced with a randomly selected pattern to maintain the population diversity of the ACO.

In detail, the judgment condition  $q \leq q_0$  determines whether to execute deterministic state probability transition rules or random selection at each iteration. Because parameter  $q$  is a uniformly distributed random number,  $q_0$  has a significant influence on the convergence performance and global search ability of the improved ACO. A small value of  $q_0$  indicates that the probability of executing the deterministic state transition rule is low and the probability of random selection

increases. Although this is beneficial for maintaining global search ability, it limits the convergence performance of the algorithm. In contrast, the convergence speed is also improved, and the diversity of the ACO is weakened. Therefore, an adaptive parameter  $q_0$  is embedded in the algorithm to balance the two path search methods. The parameter  $q_0$  is expressed in Eq. (10).

$$q_0 = \begin{cases} \frac{K-k}{K} q_{0\text{initial}}, & (k < nK) \\ \frac{(2k - 0.4K)}{2} q_{0\text{initial}}, & (k \geq nK) \end{cases} \quad (10)$$

where  $q_{0\text{initial}}$  represents the initial value of  $q_0$ , the current iteration is presented as  $k$ ,  $K$  is the maximum number of iterations, and  $nK = k_0 (n=0.7)$  is the parameter used to measure the iteration process. It can be observed from Eq. (10), when  $k < k_0$ , the iteration process is considered to be in an early stage, and a larger value of  $q_0$  implies that the search efficiency of the algorithm is improved. As the number of iterations increased, the value of  $q_0$  decreased continuously, and the global search ability and population diversity of the ACO were enhanced.

### 3.4. Non-uniform pheromone initialization

In the initialization phase of the original ACO, pheromones are uniformly distributed in the grid model, which can cause individual ants to blindly search for paths and perform unnecessary calculations, thereby reducing the convergence speed. In this study, a non-uniform pheromone initialization mechanism is designed to enhance the initialization performance of the algorithm, which enhances the ability of ACO to select advantageous regions for robot path planning. The proposed non-uniform pheromone initialization expression is given by Eq. (11).

$$\begin{cases} \tau_{ij}(\text{initial}) = \frac{d_{min}}{d_{Si} + d_{iT}} C(0), \text{freegrids} \\ \tau_{ij}(\text{initial}) = \frac{0}{d_{Si} + d_{iT}}, \text{obstaclegrids} \end{cases} \quad (11)$$

where  $d_{Si}$  and  $d_{iT}$  are the Euclidean distance of two nodes.  $d_{ST}$  denotes

the distance of  $S$  to  $T$ . The distance of  $S$  to the current node  $i$  is presented as  $d_{Si}$ .  $d_{iT}$  is the distance of current node  $i$  to the  $T$ . The value of  $C(0)$  is 1, which is the initialization pheromone concentration in original ACO.

The introduced non-uniform pheromone initialization strategy enables individual ants to better identify advantageous grids compared with the original ACO. From Fig. 4, it can be observed that the distance  $d_{ST}$  is the minimum value of the sum of the distances  $d_{Si}$  and  $d_{iT}$ , which illustrates that the pheromone initialized in the direction of  $S$  to  $T$  has the maximum value of 1 ( $C(0)$ ). Simultaneously, it can be concluded that the closer the current node is to the position of  $d_{ST}$ , the greater the value of the initialization pheromones assigned to this grid. Therefore, the non-uniform pheromone initialization strategy can make individual search paths more efficient and purposeful than traditional ACO.

To realize the escape from the local optima because of the high pheromone concentration in the advantageous grid and low pheromone concentration in the disadvantageous grid, maximum and minimum pheromone concentration limits are necessary to improve the ACO. Based on the MMAS, after updating the concentration of the pheromone  $\tau_{ij}(t)$ , its range is maintained within  $[\tau_{min}, \tau_{max}]$ . As shown in Eq. (12).

$$\tau_{min} \leq \tau_{ij}(t) \leq \tau_{max} \quad (12)$$

where  $\tau_{ij}(t)$  represents the pheromone concentration of Node  $i$  to Node  $j$  in  $t$ -th iteration, and the maximum and minimum pheromone concentrations are represented as  $\tau_{max}$  and  $\tau_{min}$ , respectively. To incorporate the previous experience of the test, an improved calculation expression for  $\tau_{max}$  and  $\tau_{min}$  is defined in Eq. (13), using the laws of the previous iterations.

$$\begin{cases} \tau_{max} = [100 + 200 * (1 - \rho)] / (1 - \rho)mpl, \\ \tau_{min} = \tau_{max} / z, \end{cases} \quad (13)$$

where  $mpl$  is the minimum path length after an ant individual completes  $t$ -th iteration,  $z=400$ . In addition, parameter  $\rho$  represents the evaporation factor of pheromone. From Eq. (13), it can be observed that the convergence efficiency of ACO will be improved by using the pheromone evaporation factor and the shortest path under the current iteration number, providing more guidance for ant individuals to select nodes. Embedding the maximum and minimum pheromone

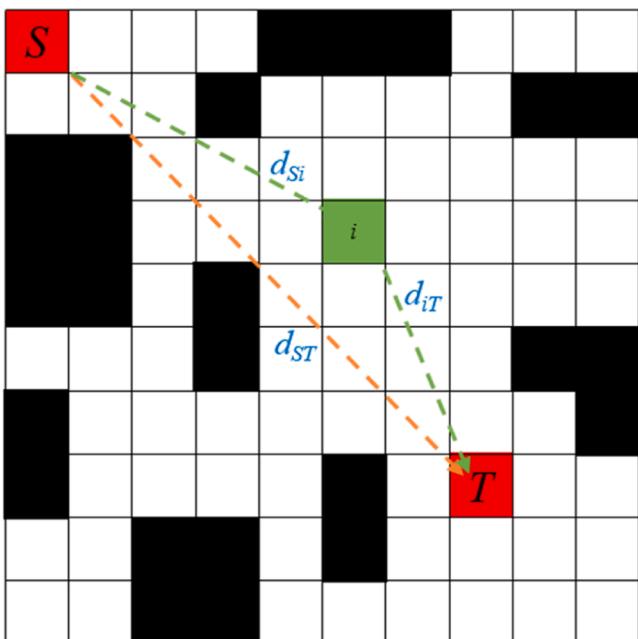


Fig. 4. Diagram of non-uniform pheromone initialization.

concentrations  $\tau_{max}$  and  $\tau_{min}$  into the algorithm can improve its global search ability.

### 3.5. The proposed MsAACO

In this study, four improvement measures were introduced, including the direction guidance mechanism, adaptive heuristic function, deterministic state transition probability rule, and nonuniform pheromone initialization combined with traditional ACO. A variant of ACO, called the multi-strategy adaptable ant colony optimization algorithm (MsAACO), was proposed for robot path planning. The pseudo-code and flowchart of the MsAACO are presented in Algorithm 1 and Fig. 5, respectively.

### 3.6. Optimization of main parameters of the proposed MsAACO

The selection of parameters is of great significance for the performance of ACO. Similarly, the environmental model also affects the performance of the algorithm. Currently, there is no comprehensive theoretical study that can directly determine the relationship between different parameter combinations and algorithm performance. Therefore, to obtain the optimal parameter combination for MsAACO, simulation experiments with different environmental models and parameter variations were statistically analyzed to determine the main MsAACO parameters. Specifically, the environmental model includes Models 1 and 2. The main parameters, including the turning factor weight  $a$ ,  $w_{hmax}$ ,  $w_{hmin}$ ,  $q_{0initial}$ , and the evaporation factor of pheromone  $\rho$  are provided in Eqs. (6), (7), (10) and (13), respectively. To maintain experimental fairness, the other parameters remained unchanged when a single parameter was tested. In addition, the experiment was repeated 10 times under the same conditions to reduce errors. The environmental model is shown in Fig. 6, where complex Model 1 has more obstacles, whereas simple Model 2 has fewer obstacles and can obtain more suitable paths. The starting node is denoted by the blue dot, and the target node is represented by green dot.

First, parameter  $a$  was tested based on Models 1 and 2. The range of the value variation for  $a$  was assigned as [1, 9], and the step size was set to 1. The values of other main parameters are as follows: the number of individuals is  $M = 50$ , the iterations is  $K = 100$ , and parameters  $\alpha$  and  $\beta$  are 1.0 and 7.0, respectively, the pheromone evaporation factor  $\rho = 0.2$ , parameter  $q_{0initial}=0.5$ , parameter  $Q = 1.0$ , parameters  $w_{hmax}$  and  $w_{hmin}$  are 0.9 and 0.2, respectively. It can be observed from Fig. 7 that the influence of  $a$  was approximately the same for the different environmental models. For turning factor weight  $a$ , the statistical results in different environmental models include the path length, turn times of the optimal path, and mean convergence generation. The black line represents the experimental results of Model 1 and the blue line represents the test results of Model 2. As shown in Fig. 7(a), the optimal path length tends to increase as the value of  $a$  changes, and a smaller  $a$  is instructive for obtaining a smaller optimal path length. From Fig. 7(b), it can be concluded that the turn times fluctuate significantly as the value of  $a$  continuously increases in the different environmental models. In addition, shorter turn times can be achieved when the value of  $a$  and the optimal path length are smaller. Therefore, a smaller value of  $a$  can help improve the performance of the algorithm. From Fig. 7(c), although waves are generated by the mean convergence generation with the alteration of  $a$ , the mean convergence generation continues to increase overall. Overall, when the value of  $a$  is set to 1, MsAACO obtains a smoother optimal path.

In adaptive heuristic function,  $H$  is the weight factor of  $d_{iT}$  and its value is determined by parameters  $w_{hmax}$  and  $w_{hmin}$ . Thus, the parameter  $w_{hmax}$  was tested based on environments Model 1 and Model 2. The range of value variation for  $w_{hmax}$  was assigned as [0.1, 0.9], and the step size was 0.1. The values of other main parameters were as follows: the turning factor weight  $a$  is 1, the number of individuals  $M = 50$ , iterations  $K = 100$ , and parameters  $\alpha$  and  $\beta$  are 1 and 7, respectively, the

**Algorithm 1**

The proposed MsAACO.

---

```

1. Employ grid method to create the grid environment model;
2. Initialize the various parameters of MsAACO, including  $K$ ,  $M$ ,  $\alpha$ ,  $\beta$ ,  $\rho$ ;
3. Initialize path route matrix,  $S$  and  $T$ ;
4. For  $n$  from 1 to  $-1$  do -1 each time
5.   For  $t$  from 1 to  $-1$  do -1 each time
6.     If  $n \neq t$  or  $n \neq 0$  then
7.       Calculate the coordinate  $(x, y)$  of the current node  $i$ ;
8.       Calculate the distance of the next node  $j$  to the  $T$ ;
9.       Calculate the distance between the last node  $i-1$  and the current node  $i$ ;
10.      If the turn times between the last node  $i-1$  and the next node  $j$  is 1 then
11.        The turning factor  $c(i)$  is 1 in  $\eta(i,j)$ ;
12.      Else
13.        The turning factor  $c(i)$  is 0 in  $\eta(i,j)$ ;
14.      End If
15.      Initialize the adaptive heuristic information matrix according to Eqs. (6),
    (7), and (8);
16.    End If
17.  End For
18. End For
19. For  $n=1$ : size of the map do
20.  For  $t = 1$ : size of the map do
21.    If the grid is free then
22.      Calculate the distance of start node  $S$  to target node  $T$ , start node  $S$  to current
        node, and current node to target node  $T$ ;
23.      Initialize non-uniform pheromone concentration matrix according to Eq.
    (11);
24.    else
25.      The pheromone concentration of the grid is 0;
26.    End If
27.  End For
28. End For
29. For  $k = 1: K$  do
30.  For  $m = 1: M$  do
31.    Place all ants at the start node  $S$ ;
32.    While the individual  $m$  does not reach the target node  $T$  do
33.      If optional points  $Length \geq 1$ ;
34.        For  $n = 1: Length$  do
35.          Take difference value  $x_1$  between horizontal ordinate of start node
            and target node;
36.          Take difference value  $y_1$  between ordinate of start node and target
            node;
37.          Take difference value  $x_2$  between horizontal ordinate of current
            node and next node;
38.          Take difference value  $y_2$  between ordinate of the current node and the
            next optional node;
39.          If  $(x_1 > 0 \text{ and } x_2 > 0) \text{ or } (x_1 < 0 \text{ and } x_2 < 0) \text{ or } (y_1 > 0 \text{ and } y_2 > 0)$ 
            or  $(y_1 < 0 \text{ and } y_2 < 0)$ 
40.          Record the nodes that satisfy adaptive heuristic function with
            orientation information;
41.        End If
42.      End For
43.      Update the array of the optional points after the heuristic mechanism with
            orientation information;
44.      Deterministic state transition probability rule;
45.      Decided  $q_{0initial}$ , calculate  $q_0$  according to Eq. (10), calculate  $k_0 = 0.7K$ ;
46.      Randomly generate  $q$  between [0,1];
47.      Select next node by Eqs. (9) and (10);
48.      Record the foraging route and route length;
49.      If all ants reach the target node  $T$  then
50.        Update the global pheromone;
51.        Calculate  $\tau_{min}$  and  $\tau_{max}$  according to Eq. (13);
52.      End If
53.    End While
54.  End For
55. End For
56. Output the final optimal path.

```

---

pheromone evaporation factor  $\rho = 0.2$ , parameter  $q_{0initial} = 0.5$ , parameter  $Q = 1$ , parameters  $w_{hmin}$  is 0.2. It can be observed from Fig. 8 that the variation of the path length, turn times, and mean convergence generation of optimal path curves in Model 1 were similar to those in Model 2. As shown in Fig. 8(a), the change of  $w_{hmax}$  has almost no effect on obtaining optimal path length in different environmental models. From

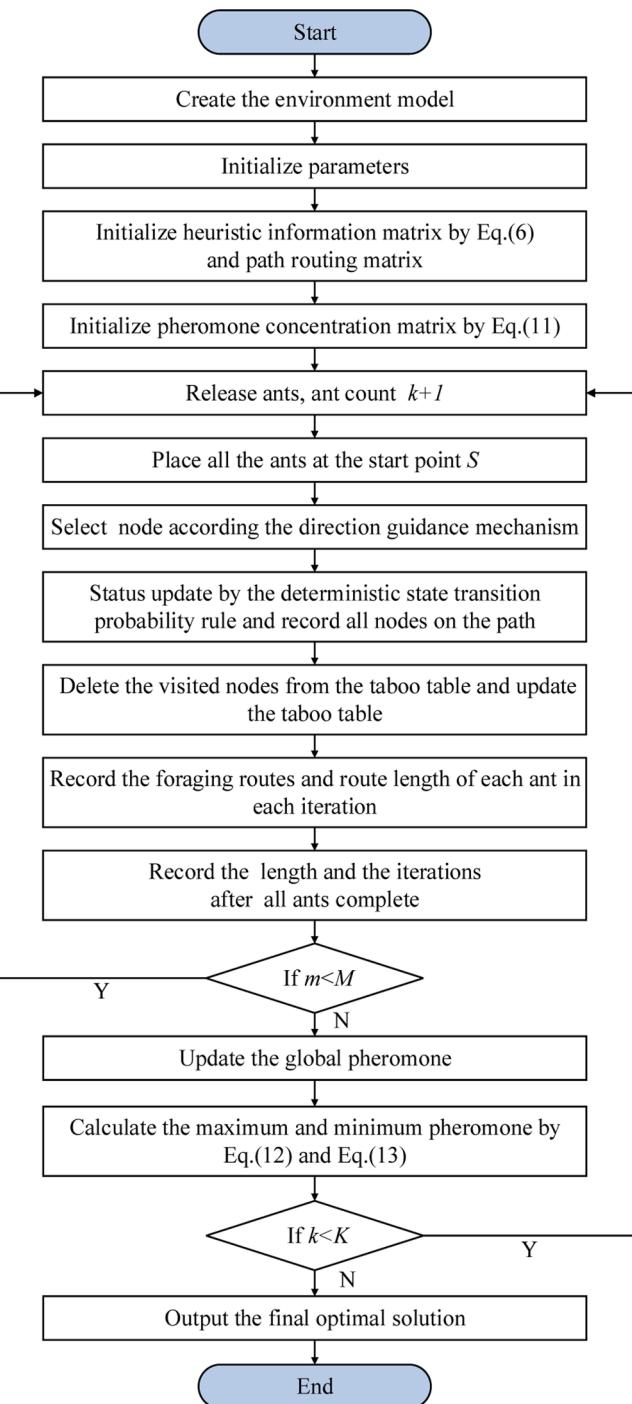
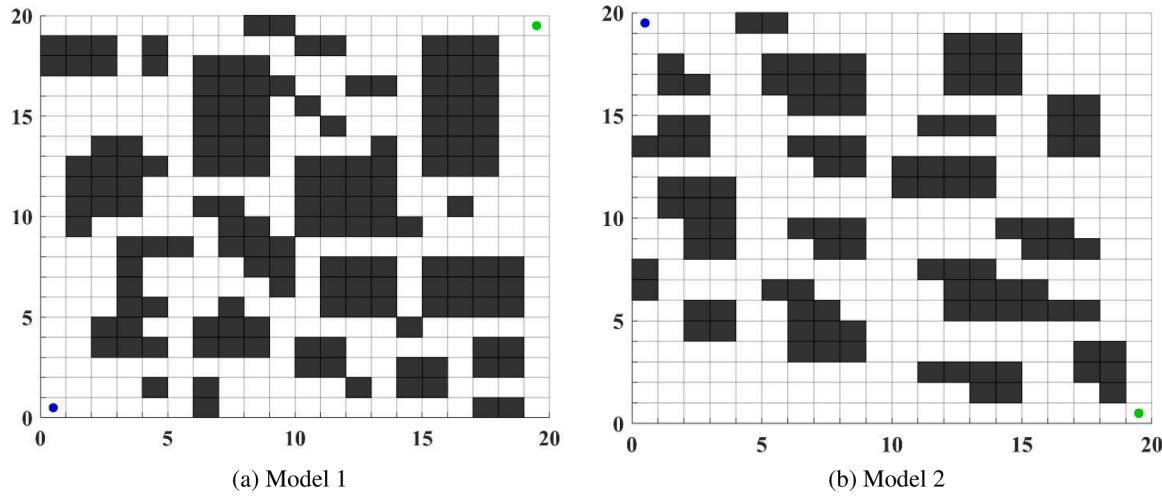


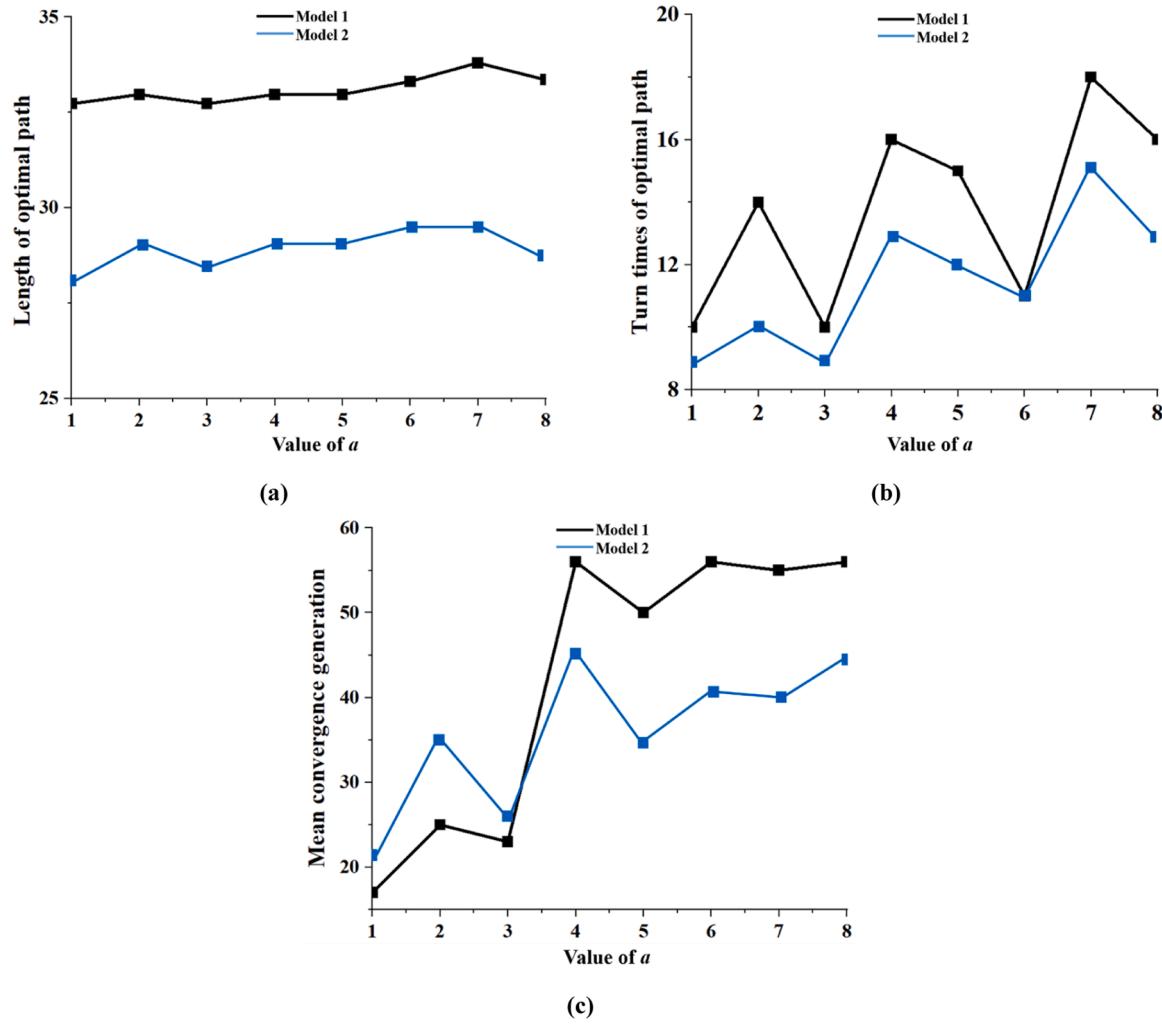
Fig. 5. Flow chart of MsAACO.

Fig. 8(b) and 8(c), parameter  $w_{hmax}$  has a significant influence on turn times and mean convergence generation. The turn times of optimal path first increased and then decreased as the value of  $w_{hmax}$  increased, ultimately achieving the minimum turn times. In addition, the mean convergence generation continuously decreases owing to the increase in the value of  $w_{hmax}$ . Therefore, the performance of the adaptive heuristic function improves as the value of  $w_{hmax}$  increases. Hence, the value of parameter  $w_{hmax}$  was set as 0.9.

As mentioned in Eq. (7), the value of  $H$  is determined by parameters  $w_{hmax}$  and  $w_{hmin}$ . Thus, the parameter  $w_{hmin}$  was tested in different environment models including Model 1 and Model 2. The range of value variation for  $w_{hmin}$  was assigned as [0.1, 0.9], and the step size was 0.1.



**Fig. 6.** Experimental environment models of parameter optimization.



**Fig. 7.** Influences of turning factor weight  $a$  on optimal path.

Similarly, the values of other main parameters were as follows: the turning factor weight  $a$  is 1, parameter  $w_{hmax}=0.9$ ,  $M = 50$ , iterations  $K = 100$ , parameters  $\alpha$  and  $\beta$  are 1 and 7, respectively, the pheromone evaporation factor  $\rho = 0.2$ , parameter  $q_{0initial}=0.5$ , parameter  $Q = 1$ . The statistical results under different conditions are displayed in Fig. 9. It can

be observed that the variation of the path length, turn times, and mean convergence generation of optimal path curves also had a similar trend in two environment models. As exhibited in Fig. 9(a), the optimal path length only has a small variation as the parameter  $w_{hmin}$  changes in different environmental models. From Fig. 9(b) and 9(c), it can be

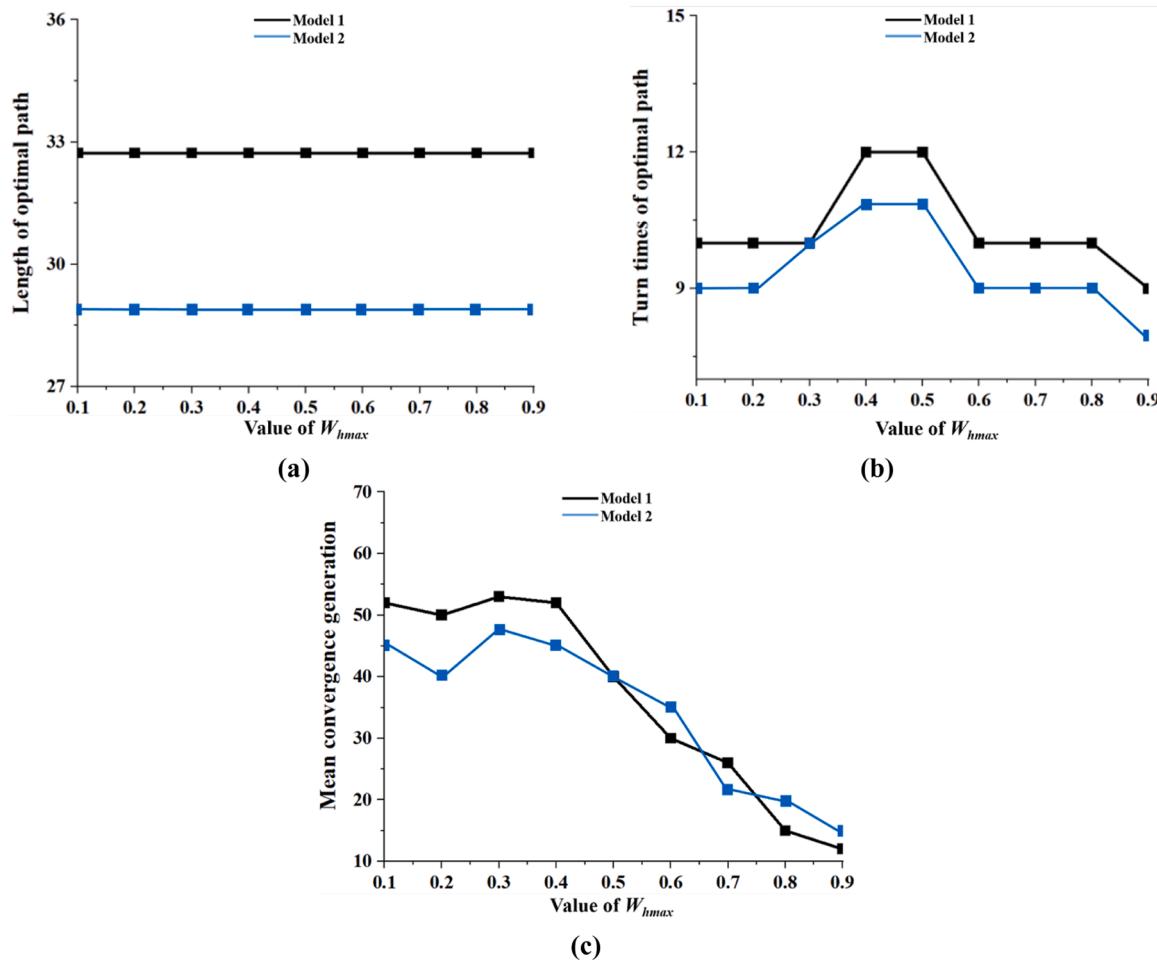


Fig. 8. Influences of parameter  $w_{hmax}$  on optimal path.

observed that both turn times and mean convergence generation have minimum values in Model 1 when the value of parameter  $w_{hmin}$  is 0.2. Moreover, in Model 2, the turn times have the minimum value when the value of  $w_{hmin}$  is 0.2 or 0.3. As  $w_{hmin}=0.2$ , the mean convergence generation has the minimum value. Therefore, the value of parameter  $w_{hmin}$  was set as 0.2 to obtain a better optimal path, which also proved that parameter  $w_{hmin}$  has a significant influence on adaptive heuristic function. In addition, the efficiency of algorithm search is effectively enhanced by introducing an adaptive heuristic function into ACO.

As mentioned in Section 3.3, the  $q_0$  is proposed to determine whether to execute deterministic state probability transition rules or random selection. The value of  $q_{0initial}$  was tested based on Model 1 and Model 2. The range of value variation for  $q_{0initial}$  was [0.1, 0.9], and the step size was 0.1. Similarly, the values of other main parameters were as follows: the turning factor weight  $a = 1$ , the value of parameters  $w_{hmax}$  and  $w_{hmin}$  was 0.9 and 0.2, respectively,  $M = 50$ ,  $K = 100$ , the value of parameters  $\alpha$  and  $\beta$  were 1 and 7, respectively, parameter  $Q = 1$ . The statistical results under different conditions are displayed in Fig. 10. For Fig. 10(a), the optimal path length only has a small fluctuation as the parameter  $q_{0initial}$  changes in different environmental models. From Fig. 10(b), the turn times have minimum values in two environment models when the value of parameter  $q_{0initial}$  is 0.5. From Fig. 10(c), the mean convergence generation has the minimum value in Model 1, when  $q_{0initial}=0.5$ , and it has the minimum value in Model 2, when  $q_{0initial}$  is in the range of [0.5, 0.6]. Therefore, the value of  $q_{0initial}$  was set as 0.5 to obtain the optimal path.

The pheromone evaporation factor  $\rho$  affects the search performance of ACO, and it is not conducive to ant individuals selecting nodes as its

value is too large or too small. Therefore, the value of parameter  $\rho$  was tested in different environmental models. Similarly, the range of value variation for  $\rho$  was [0.1, 0.9], and the step size was 0.1. The values of other main parameters were as follows:  $a = 1$ , the values of parameters  $w_{hmax}$  and  $w_{hmin}$  was 0.9 and 0.2, respectively,  $q_{0initial}=0.5$ ,  $M = 50$ ,  $K = 100$ , the value of parameters  $\alpha$  and  $\beta$  are 1 and 7, respectively, parameter  $Q = 1$ . The statistical experimental results, including the path length, turn times, and mean convergence generation of the optimal path, are shown in Fig. 11. From Fig. 11(a), the alteration of parameter  $\rho$  has a small effect on obtaining the optimal path length in different environmental models. From Fig. 11(b) and 11(c), the turn times and mean convergence generation have minimum values in two environment models as the value of parameter  $\rho$  is 0.2. Therefore, the value of parameter  $\rho$  was set as 0.2.

After sensitivity analysis of parameters in different environmental models, the optimal values of the main parameters were assigned as follows:  $M = 50$ ,  $K = 100$ , and parameters  $\alpha$  and  $\beta$  are 1 and 7, respectively, the pheromone evaporation factor  $\rho = 0.2$ , parameter  $q_{0initial}=0.5$ , parameter  $Q = 1$ , parameters  $w_{hmax}$  and  $w_{hmin}$  were 0.9 and 0.2. By introducing the optimal parameter values into MsAACO, the optimal path and convergence curve for Model 1 were obtained, as shown in Fig. 12(a) and 12(b), respectively. The turn times is 8, the number of iterations is 7 and the optimal path length is 32.73. The optimal path and convergence curve generated in environment Model 2 are displayed in Fig. 13(a) and 13(b); the turn times is 7, the number of iterations is 7 and the optimal path length is 28.63.

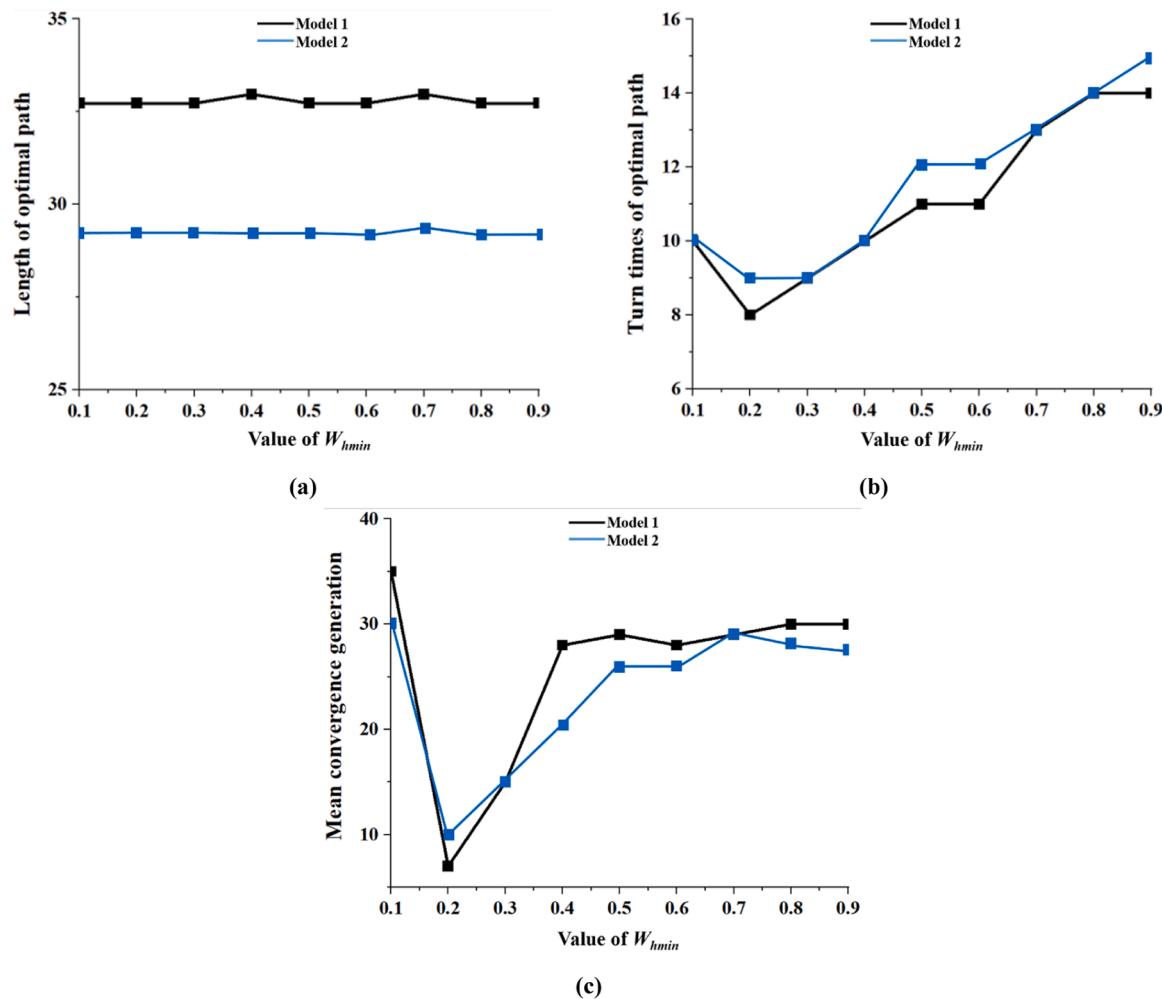


Fig. 9. Influences of parameter  $w_{hmin}$  on optimal path.

#### 4. Experiments and numerical analysis

In this section, experiments were conducted based on different environments to demonstrate the superiority of the proposed MsAACO. Comprehensive experiments and analyses were conducted to verify the effectiveness of the four strategies. In addition, five environmental models and several current advanced algorithms, including the A\* algorithm, variants of ACO, Dijkstra's algorithm, JPS, best-first search algorithm, breadth-first search algorithm, trace algorithm, and existing excellent algorithms, were employed to validate the outstanding performance of MsAACO in solving robot path-planning problems.

##### 4.1. Effectiveness verification of the four proposed strategies

In this section, four proposed strategies, namely, the direction guidance mechanism, nonuniform pheromone initialization, adaptive heuristic function, and deterministic state transition probability rule, were tested and analyzed to verify their effectiveness.

In addition, the four proposed strategies were combined with the original ACO and three transitional variants of ACO, ACO<sub>1</sub>, ACO<sub>2</sub>, and ACO<sub>3</sub>, which are listed in Table 1. Five algorithms—ACO, ACO<sub>1</sub>, ACO<sub>2</sub>, ACO<sub>3</sub>, and MsAACO—were operated under the same experimental conditions. The grid environmental model ( $20 \times 20$ ) is shown in Fig. 14, where the starting node is represented by a blue dot, and the target node is represented by green dot. The values of main parameters are:  $M = 50$ ,  $K = 100$ ,  $\rho = 0.2$ ,  $q_{0initial} = 0.5$ ,  $\alpha = 1.0$ ,  $\beta = 7.0$ ,  $Q = 1.0$ ,  $w_{hmax} = 0.9$ ,  $w_{hmin} = 0.2$ . Moreover, to reduce randomness errors, each algorithm was

repeatedly run ten times in a path-planning environment model. The optimal path and mean convergence generation are displayed in Figs. 14 and 15. The statistical results are listed in Table 2.

As shown in Table 1, the effectiveness of the four strategies was verified by sequentially embedding the four improvement strategies into the traditional ACO. The statistical results presented in Table 2 show that, compared with traditional ACO, ACO<sub>1</sub> with the introduction of directional guidance mechanism has a shorter optimal path length ( $32.3248 < 34.0416$ ), shorter mean of path length ( $32.3248 < 34.6274$ ), fewer turn times ( $8 < 11$ ), and less mean convergence generation ( $50 < 74$ ). The results indicate that the direction-guidance mechanism effectively improves the directionality of the ant individuals when selecting nodes and enhances the search efficiency of the ACO.

Moreover, the performance of ACO<sub>2</sub> is improved by introducing an adaptive heuristic function. ACO<sub>2</sub> has a shorter path length ( $31.5563 < 32.3248$ ), fewer turns ( $8 < 11$ ), and less mean convergence generation ( $27 < 50$ ) than ACO<sub>1</sub>, demonstrating that the location information and turn times embedded in the adaptive heuristic function can effectively enhance the accuracy of ACO.

Compared to ACO<sub>2</sub>, ACO<sub>3</sub> imports a deterministic state transition rule based on the embedding of adaptive heuristic functions. From Table 2, although ACO<sub>3</sub> has a slight deviation in the mean convergence generation ( $27 > 22$ ), it ACO<sub>3</sub> has advantages in terms of turn times ( $6 < 7$ ), demonstrating that the deterministic state transition rule may weaken the convergence of ACO, while improving its path-planning performance.

After adding the nonuniform pheromone initialization strategy in

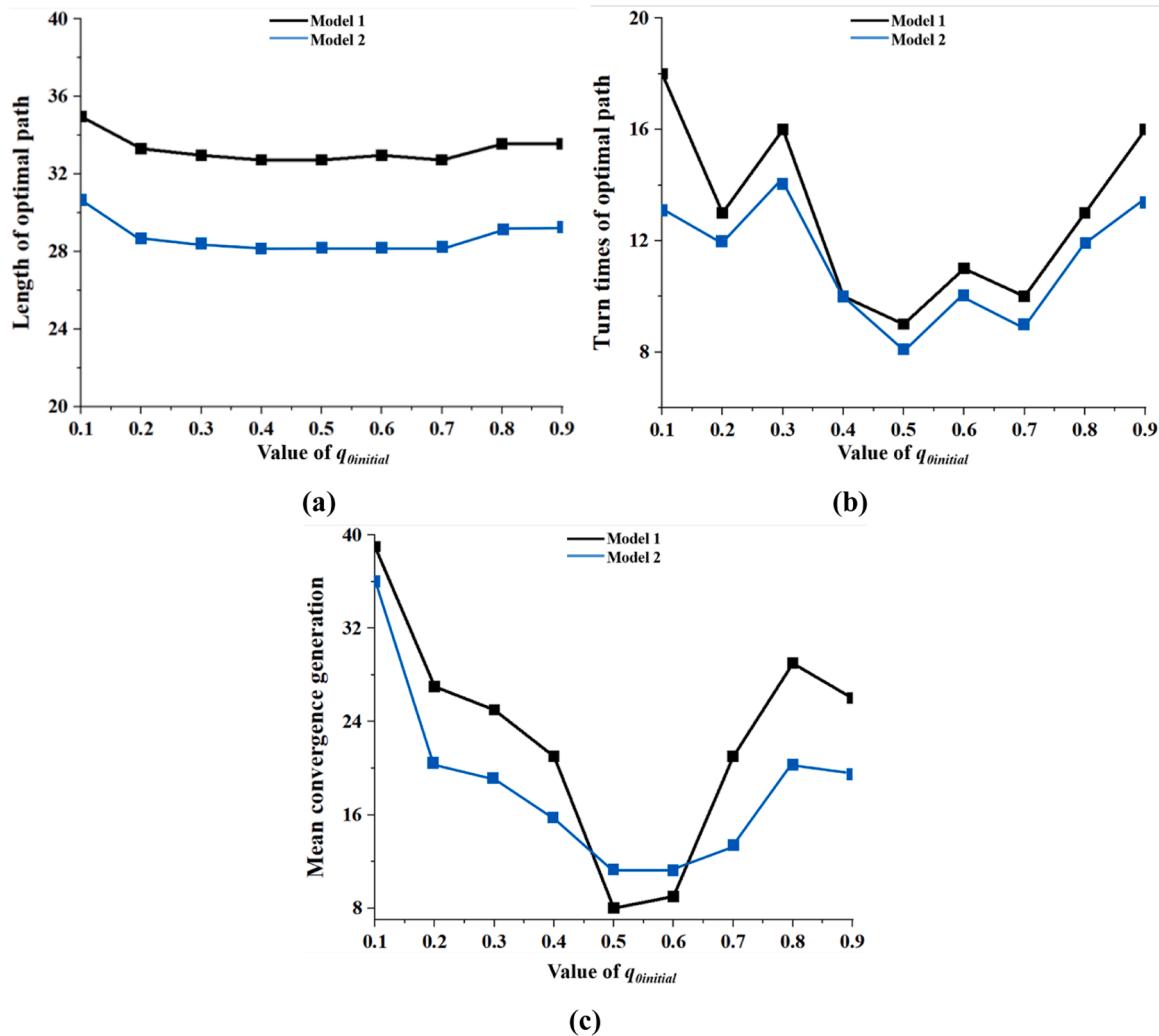


Fig. 10. Influences of parameter  $q_{0\text{initial}}$  on optimal path.

ACO<sub>3</sub>, MsAACO displayed excellent performance in path planning. From Table 2, it can be observed that MsAACO obtained the shortest optimal path length (31.5563), smallest turn times (4) in the optimal path, and lowest mean convergence generation (15), which verifies that nonuniform pheromone initialization enhances the search purposefulness and improves the search efficiency of the algorithm.

After the introduction of new strategies into the transition variants of ACO, the optimal paths obtained by ACO, ACO<sub>1</sub>, ACO<sub>2</sub>, ACO<sub>3</sub>, and MsAACO continuously improved. The convergence curves further verify the effectiveness of the four strategies. Overall, the four strategies, including the direction guidance mechanism, adaptive heuristic function, deterministic state transition probability rule, and nonuniform pheromone initialization, improved the path-planning performance of the algorithm and had outstanding effects.

#### 4.2. Simulation experiment I

First, the proposed MsAACO is compared with a series of excellent algorithms involving the original ACO, two variants of ACO named the heuristic improvement ant colony algorithm (IACO), adaptive ant colony algorithm (IAACO) [54], Dijkstra's algorithm, and A\* algorithm, to verify its remarkable performance. To ensure the fairness of the experiment, the grid environment model was utilized in [54], as displayed in Fig. 16. The map size of the grid environment model was 20 × 20, with coordinates (0.5, 19.5) for the starting node, represented by the red dot, and the target node with the coordinates (19.5, 0.5), represented by the

blue dot. White grids represent free space and black grids denote obstacles. All experiments were independently performed 20 times to avoid random errors [55].

As shown in Table 3, the common parameters of ACO and three variants of ACO, MsAACO, IAACO, and IACO, were assigned the same values in this simulation experiment. Moreover, the parameters of three variants of ACO are set as follows:  $M = 50$ ,  $K = 100$ ,  $k = 0.9$ ,  $\alpha = 1.0$ ,  $\beta = 7.0$ ,  $Q = 2.5$ ,  $\rho = 0.2$ ,  $\sigma_1 = 0.1$ ,  $\sigma_2 = 0.9$ ,  $\delta_0 = 0.15$ ,  $\lambda = 7.0$ ,  $K_S = 0.1$ ,  $K_E = 0.2$ ,  $R_s = 0.5$ , and  $k_L = 0.7$ , in which the specific parameters including weight coefficient  $\lambda$ , adjustment coefficient  $k$ , distance weight coefficients  $\sigma_1$  and  $\sigma_2$  belong to IACO, IAACO, and ACO, respectively. Detailed descriptions can be found in Reference [54]. The specific parameters of MsAACO are assigned as follows: the pheromone evaporation factor  $\rho = 0.2$ , parameter  $q_{0\text{initial}} = 0.5$ , parameter  $a = 1$ , parameters  $w_{h\text{max}}$  and  $w_{h\text{min}}$  are 0.9 and 0.2, respectively.

In addition, the proposed MsAACO was compared with the Dijkstra's and A\* algorithms to verify its superiority in path planning. The A\* algorithm is a heuristic search algorithm that uses an evaluation function to estimate the benefits of selecting each node, such that the search individuals reach the nodes that appear closer to the target. Dijkstra's algorithm is a popular algorithm that utilizes graph theory to determine the shortest distance to the next node during path planning. The statistical results of the obtained optimal paths for all algorithms are listed in Table 4. In addition, the optimal paths for solving the environmental models are shown in Figs. 16 and 17. The optimal convergence curves for the four ACOs are presented in Fig. 18.

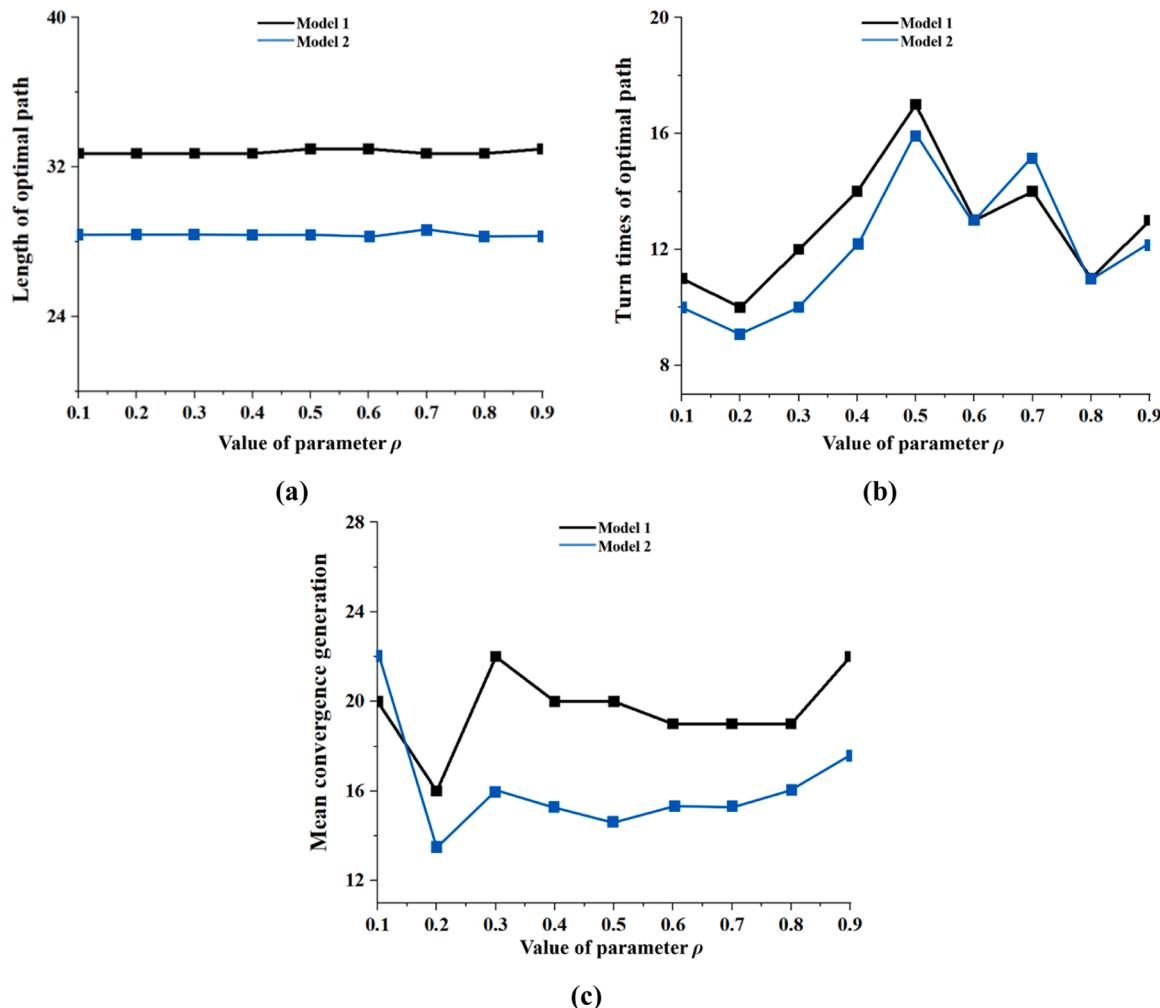
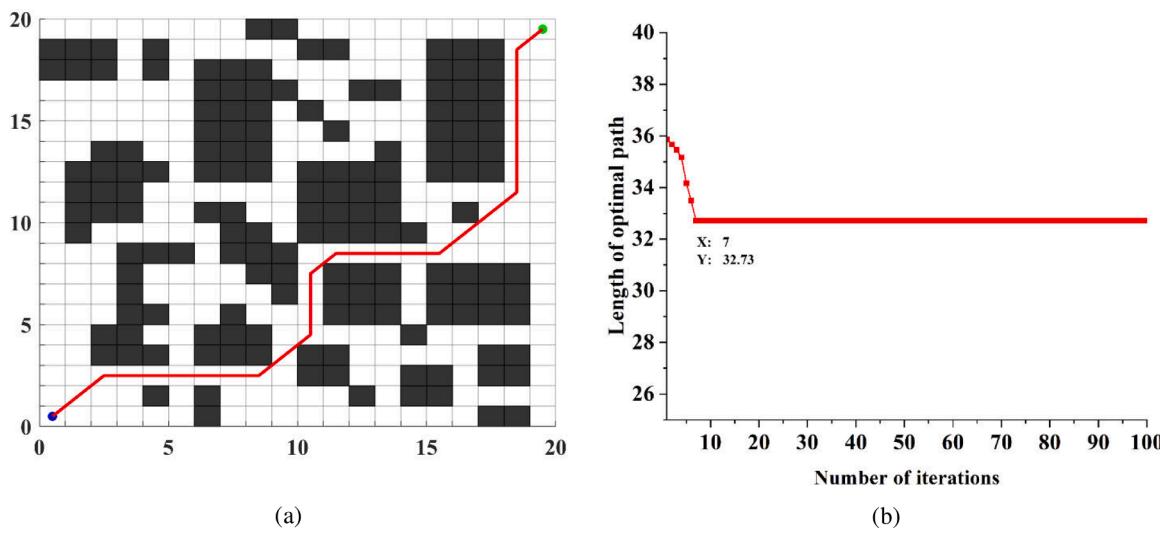
Fig. 11. Influences of pheromone evaporation factor  $\rho$  on optimal path.

Fig. 12. Optimal path and convergence curve for environment Model 1.

It can be observed from Table 4, shortest length of optimal path (28.63) obtained using MsAACO is equal to that of A\* algorithm and IAACO, which is smaller than optimal path lengths generated by the ACO, IACO and Dijkstra's algorithm ( $28.63 < 29.80 < 31.80 < 32.14$ ). In

addition, the MsAACO has the lowest mean path length and standard deviation (Std.). Moreover, the “Mean” of path length generated by the MsAACO is lower than that of IAACO, Dijkstra's algorithm, IACO, and ACO ( $28.63 < 29.51 < 29.80 < 33.26 < 33.76$ ), and Std. of MsAACO was

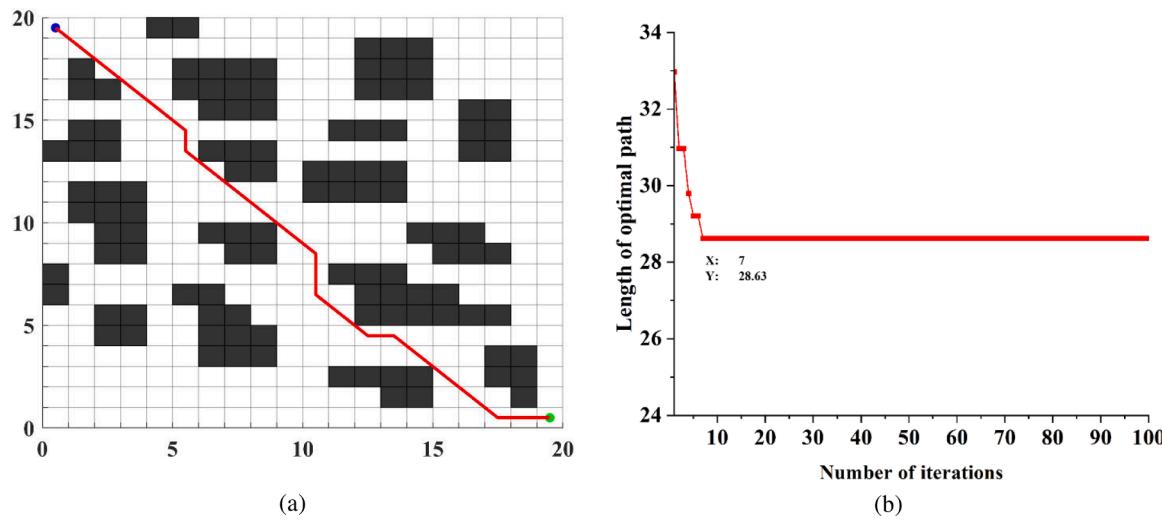


Fig. 13. Optimal path and convergence curve for environment Model 2.

**Table 1**  
Different variants of ACO combined with different strategies.

Different transitional ACOs	Abbreviation
Original ACO	ACO
ACO with direction guidance mechanism (the strategy proposed in Section 3.1)	ACO <sub>1</sub>
ACO with direction guidance mechanism and adaptive heuristic function (the strategies proposed in Section 3.1 and Section 3.2)	ACO <sub>2</sub>
ACO with direction guidance mechanism, adaptive heuristic function, and deterministic state transition probability rule (the strategies proposed in Section 3.1, Section 3.2, and Section 3.3)	ACO <sub>3</sub>
ACO with all the four strategies proposed	MsAACO

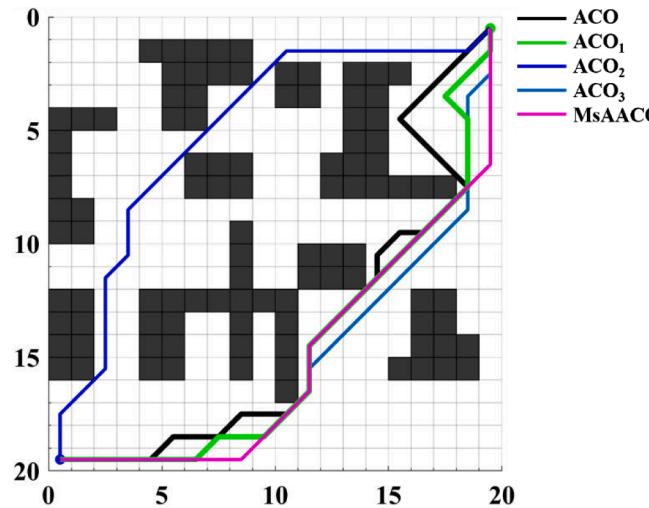


Fig. 14. Generated optimal paths of five ACOs.

lower than that of IAACO, IACO, and ACO ( $0 < 0.75 < 0.83 < 2.93$ ). The statistical results of the path length show that the proposed MsAACO has strong search stability and superior path length.

In addition, the proposed MsAACO algorithm is superior to the other algorithms in terms of turn time. In detail, compared with ACO, Dijkstra's algorithm, A\* algorithm and IACO, the "Turn times" generated by MsAACO are better on the optimal path ( $7 < 9 = 9 < 16$ ), and same as that of IAACO. As for the number of convergence generation, from

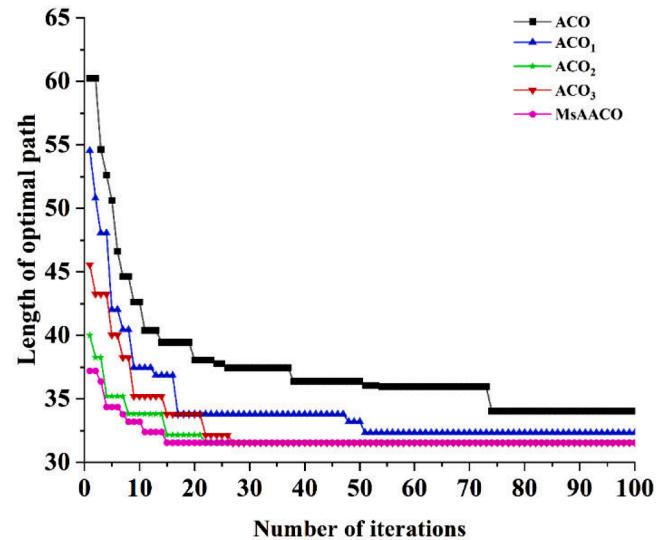


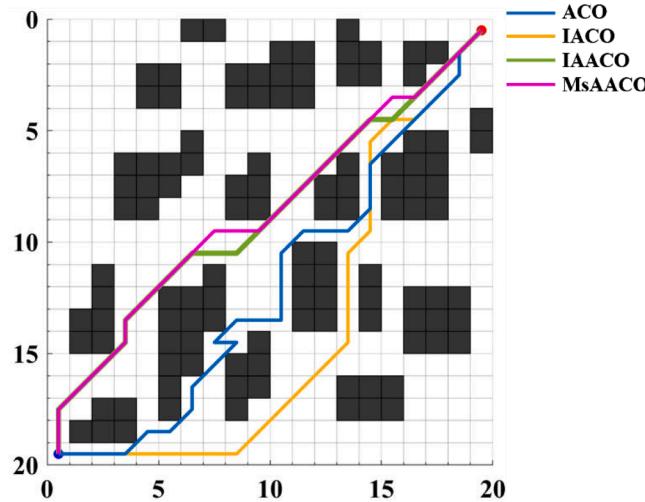
Fig. 15. Convergence curves of five ACOs.

**Table 2**  
Statistical results of five ACOs.

	Optimal path length	Mean of path length	Turn times	Mean of convergence generation
ACO	34.0416	34.6274	11	74
ACO <sub>1</sub>	32.3248	32.3248	8	50
ACO <sub>2</sub>	31.5563	31.5563	7	22
ACO <sub>3</sub>	31.5563	31.5563	6	27
MsAACO	31.5563	31.5563	4	15

**Table 4** and **Fig. 18**, the proposed MsAACO produces a smaller "Best" ( $5 < 6 < 17$ ) and "Mean" ( $5.3 < 9.3 < 27.4$ ) of convergence generation than that of ACO and IACO, while has a slight deviation from the results of IAACO. Furthermore, it can be found that the MsAACO occupies an evident advantage at the level of "Std." of convergence generation (0.2202), indicating that the MsAACO has excellent stability in solving path-planning problems.

**Figs 16 and 17** also present the optimal paths of MsAACO, Dijkstra's algorithm, A\* algorithm, and variants of ACO. The optimal path obtained using MsAACO had fewer turn times and path lengths than those of the other algorithms. Thus, benefiting from the proposed direction



**Fig. 16.** Optimal paths obtained using different ACOS.

guidance mechanism, adaptive heuristic function, deterministic state transition probability rule, and nonuniform pheromone initialization, MsAACO is the best choice for handling path-planning problems. Higher optimal path quality stability is the core advantage of MsAACO compared to other algorithms for the testing environment model. Additional path planning algorithms and environmental models were employed to further validate the effectiveness of MsAACO in path-planning problems.

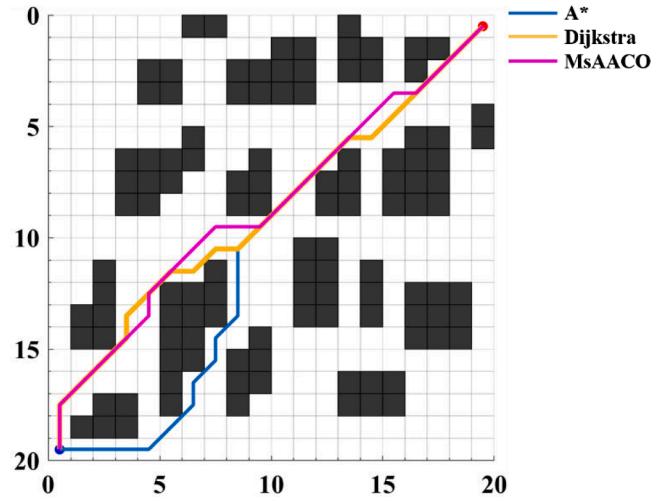
#### 4.3. Simulation experiment II

In this section, the proposed MsAACO is compared with three state-of-the-art algorithms: PS-ACO [56], an improved ACO called AIACSE [57], and an adaptive ACO named MRCACO [58] in a complex environment model mentioned in the literature [57]. Furthermore, the Dijkstra's algorithm, A\* algorithm, and original ACO were employed to evaluate the performance of MsAACO in path planning. Similarly, to ensure the fairness of the experiment, the grid environment model in [57] was utilized, as shown in Fig. 19. The size of the grid-environment model was  $30 \times 30$ . The coordinates of the starting node are (0.5, 29.5), and are represented by a red dot. The coordinates of the target node, which is represented by a blue dot, are (29.5, 0.5). White grids represent free space and black grids represent obstacles. All experiments were independently performed 20 times to avoid random errors.

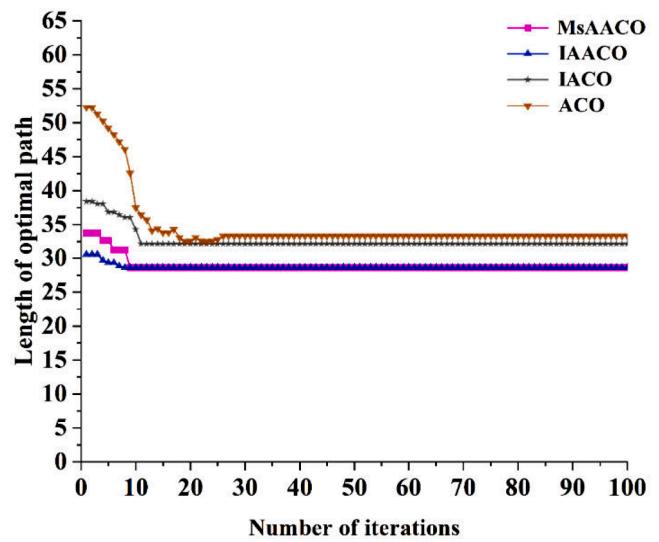
To maintain the original performance of the compared algorithms, the parameter settings of the PS-ACO, AIACSE, and MRCACO mentioned in [57] were consistent with those in the literature. In addition, the common parameter settings of MsAACO and the original ACO were the same as those of AIACSE to maintain the fairness of the experiment. As shown in Table 5, the common parameters of variants ACO are the same as those of AIACSE and assigned as follows: the number of individuals  $M = 60.0$ , iterations  $K = 200.0$ ,  $\alpha=1.1$ ,  $\beta=7.0$ ,  $\rho=0.2$ ,  $Q = 100.0$ , and  $\rho=0.2$ . The specific parameters  $q_{0\min}$  and  $q_{0\max}$  of AIACSE are 0.4 and 0.9, respectively. The specific parameters of MsAACO are as follows:  $q_{0\text{initial}}=0.5$ ,  $w_{h\max}=0.9$ ,  $w_{h\min}=0.2$ , and  $a = 1.0$ . The parameters of PSO-ACO are set the same as in literature [56], which are  $M = 2000$ ,  $K =$

**Table 4**  
Experimental results of different algorithms.

Algorithms	Path length			Convergence generation			Turn times
	Best	Mean	Std.	Best	Mean	Std.	
ACO	31.80	33.76	2.93	17.00	27.42	5.54	16.00
IACO	32.14	33.26	0.83	6.00	9.30	3.61	9.00
Dijkstra's	29.80	29.80	0.00	—	—	—	9.00
A*	28.63	28.63	0.00	—	—	—	9.00
IAACO	28.63	29.51	0.75	3.00	4.00	0.89	7.00
MsAACO	28.63	28.63	0.00	5.00	5.33	0.22	7.00



**Fig. 17.** Optimal paths obtained using MsAACO, A\*, and Dijkstra's algorithms.



**Fig. 18.** Convergence curves of different ACOS.

**Table 3**  
Parameter settings in this experiment.

Algorithms	$M$	$K$	$k$	$\alpha$	$\beta$	$Q$	$\rho$	$q_{0\text{initial}}$	$\alpha$	$w_{h\max}$	$w_{h\min}$	$\sigma_1$	$\sigma_2$	$\delta_0$	$\lambda$	$K_S$	$K_E$	$R_s$	$k_L$
ACO	50	100	0.90	1.0	7.0	2.5	0.2	—	—	—	—	0.1	0.9	0.15	7.0	0.1	0.2	0.5	0.7
IAACO	50	100	0.90	1.0	7.0	2.5	—	—	—	—	—	0.1	0.9	0.15	7.0	0.1	0.2	0.5	0.7
IACO	50	100	0.90	1.0	7.0	2.5	—	—	—	—	—	0.1	0.9	0.15	7.0	0.1	0.2	0.5	0.7
MsAACO	50	100	—	1.0	7.0	2.5	0.2	0.5	1.0	0.9	0.2	—	—	—	—	—	—	—	—

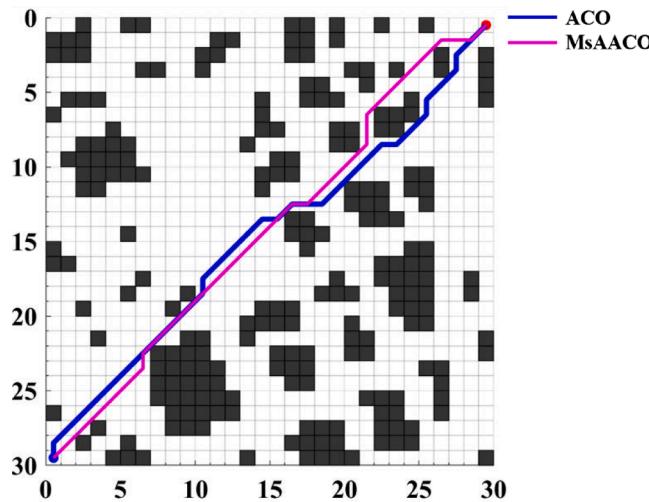


Fig. 19. Optimal paths obtained using MsAACO and ACO.

2000,  $\alpha=1.0$ ,  $\beta=2.0$ ,  $Q=100.0$ , and  $\rho=0.015$ . Similarly, the parameters mentioned in MRCACO are set as follows:  $M=60$ ,  $K=2000$ ,  $\alpha=1.0$ ,  $\beta=4.0$ ,  $Q=1.0$ , and  $\rho=0.1$ . The statistical results obtained using the ACO, Dijkstra, and A\* algorithms are listed in Table 6. The optimal paths generated by these algorithms when solving the environmental models are shown in Figs. 19–24.

As can be observed from experimental results, the proposed MsAACO has a certain superiority over the other six comparison algorithms in terms of “Path length” and “Turn times” for this experiment. Through the “Best” and “Mean” of path length generated by the MsAACO, which is equal to MRCACO, A\* algorithm and Dijkstra algorithm (42.77), and less than those of PS-ACO, AIACSE and ACO, the MsAACO gets the minimum “Turn times” with a value of 7. Moreover, the “Turn times” of MsAACO is far less than that of ACO (13), PS-ACO (11), MRCACO (9), AIACSE (17), A\* algorithm (9), and Dijkstra algorithm (9), which indicates that MsAACO can effectively reduce the energy consumption of robot turns. The “Path length” and “Turn times” of MsAACO generally exceed those of other six comparison algorithms for this environment model. Benefiting from these four proposed improvements, MsAACO has remarkable stability and superiority in solving path-planning problems. Figs. 19–24 exhibit the optimal path solutions of MsAACO and various comparison algorithms, in which MsAACO has significant advantages in path planning and can obtain a better optimal path solution than those algorithms.

#### 4.4. Simulation experiment III

In this section, to demonstrate the superiority of the proposed MsAACO, it is compared with an existing algorithm in the literature [59], the original ACO, Dijkstra’s, and A\* algorithms in the space environment model [59]. It can be observed from Fig. 25 that the grid environment model size is  $20 \times 20$ . The starting and target nodes are represented by red and blue dots, respectively, and the coordinates of the starting and target nodes are (0.5,19.5) and (19.5, 0.5), respectively. To avoid random errors, the common parameters of the MsAACO and

**Table 6**  
Experimental results of different algorithms.

Algorithms	Path length			Turn times
	Best	Mean	Std.	
AIACSE	44.53	44.53	0	17
ACO	43.36	47.94	1.32	13
PS-ACO	43.36	45.42	1.79	11
MRCACO	<b>42.77</b>	<b>42.77</b>	0	9
Dijkstra	<b>42.77</b>	<b>42.77</b>	0	9
A*	<b>42.77</b>	<b>42.77</b>	0	9
MsAACO	<b>42.77</b>	<b>42.77</b>	0	7

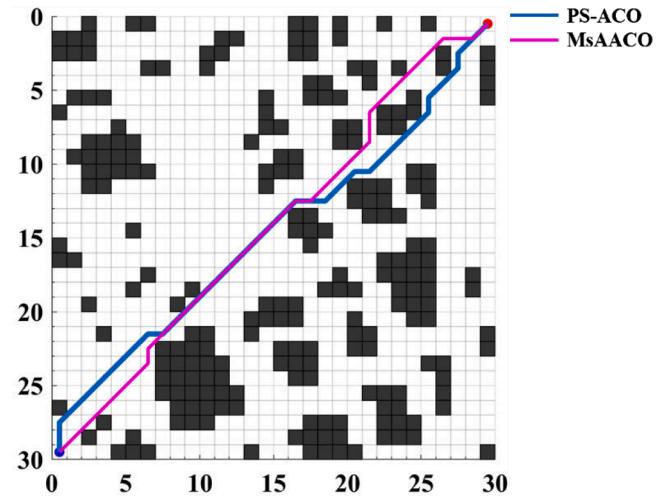


Fig. 20. Optimal paths obtained using MsAACO and PS-ACO.

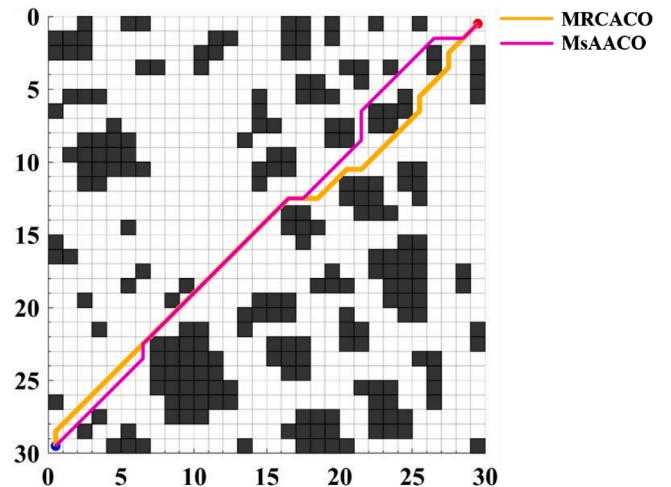


Fig. 21. Optimal paths obtained using MsAACO and MRCACO.

**Table 5**  
Parameter settings in this experiment.

Algorithms	$M$	$K$	$\alpha$	$\beta$	$Q$	$\rho$	$q_{0\text{initial}}$	$w_{h\text{max}}$	$w_{h\text{min}}$	$a$	$q_{0\text{min}}$	$q_{0\text{max}}$
ACO	60.0	200.0	1.1	7.0	100.0	0.2	–	–	–	–	–	–
AIACSE	60.0	200.0	1.1	7.0	100.0	0.2	–	–	–	–	0.4	0.9
MsAACO	60.0	200.0	1.1	7.0	100.0	0.2	0.5	0.9	0.2	1.0	–	–
PS-ACO	2000	2000	1.0	2.0	100.0	0.015	–	–	–	–	–	–
MRCACO	60.0	2000	1.0	4.0	1.0	0.1	–	–	–	–	–	–

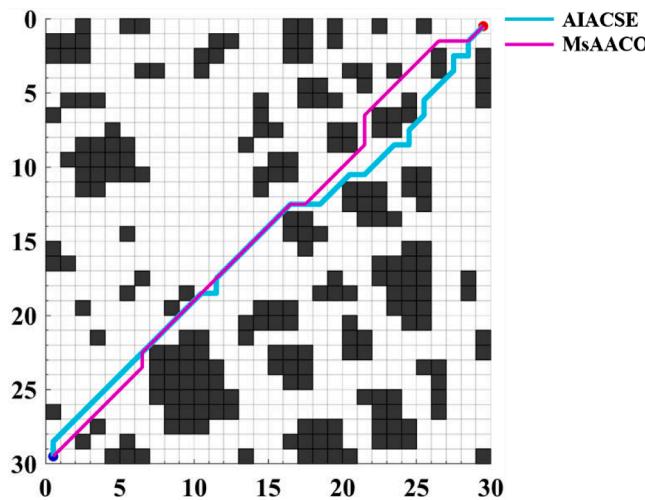


Fig. 22. Optimal paths obtained using MsAACO and AIACSE.

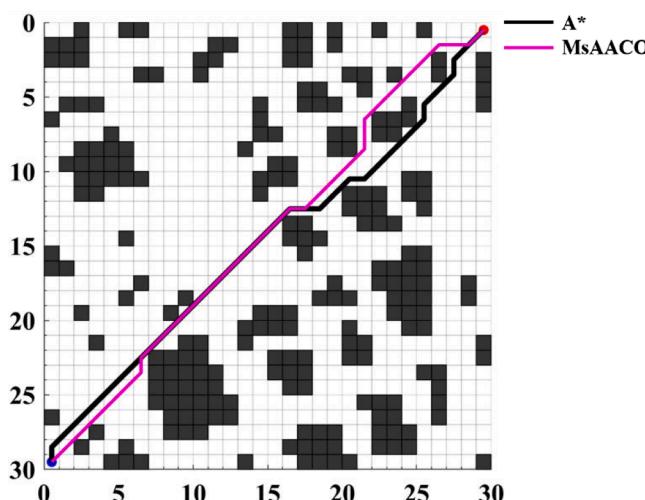


Fig. 23. Optimal paths obtained using MsAACO and A\* algorithm.

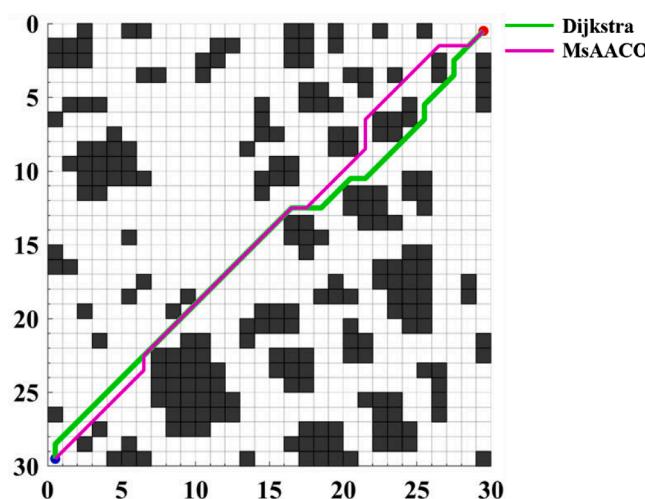


Fig. 24. Optimal paths obtained using MsAACO and Dijkstra algorithm.

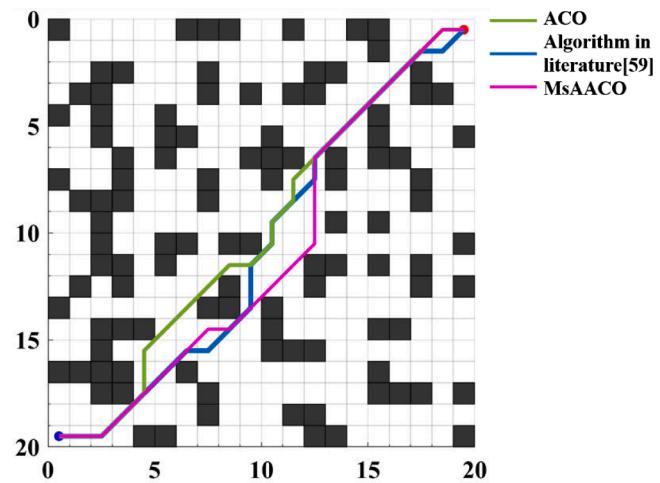


Fig. 25. Optimal paths obtained using MsAACO, the algorithm in Reference [59], and ACO.

the original ACO were arranged in the same manner as the algorithm in the literature [59]. All experiments were independently tested 20 times. The statistical results for the ACO, MsAACO, Dijkstra's algorithm, A\*, and the algorithm in literature [59] are listed in Table 7. The optimal paths generated by these algorithms are shown in Figs. 25–26, and the optimal convergence curves of the MsAACO, ACO, and the algorithm proposed in Reference [59] are shown in Fig. 27.

As shown in Table 7, the proposed MsAACO has a certain advantage over the other path-planning comparison algorithms in this experiment. Through the “Best” path length generated by the MsAACO, which is equal to ACO, A\* algorithm and the algorithm in Reference [59] (29.21), and less than that of Dijkstra's algorithm (30.04), the MsAACO obtains the minimum turn times with a value of 7. Moreover, the “Turn times” of MsAACO is far less than that of ACO (11), the algorithm in Reference [59] (10), A\* algorithm (9), and Dijkstra's algorithm (9). As shown in Fig. 25–26, the path-planning solution of MsAACO is smoother with the least turn times compared with the other algorithms, which indicates that MsAACO can effectively reduce the energy consumption of mobile robots. Fig. 25 also confirms that MsAACO has a better convergence speed than ACO, which displays the same performance as the algorithm proposed in Reference [59]. From Fig. 27 and Table 7, although the proposed MsAACO has a small deviation in convergence curve compared to the algorithm in Reference [59] ( $9 > 5$ ), Table 7 shows that the proposed MsAACO is more stable because the “Mean” of convergence iterations is 9.8. Thus, the statistical results prove that MsAACO is significantly superior in solving path-planning problems.

#### 4.5. Simulation experiment IV

In this section, the proposed MsAACO is compared with an existing algorithm in Reference [60], a variant of ACO called APFACO proposed in Reference [61], the original ACO, the Dijkstra's algorithm, and A\*

**Table 7**  
Experimental results of different algorithms.

Algorithms	Path length			Iterations			Turn times
	Best	Mean	Std.	Minimum	Mean	Std.	
Dijkstra	30.04	30.04	0.0	–	–	–	9
ACO	<b>29.21</b>	33.55	1.58	49	54.35	2.26	11
Algorithm in literature [59]	<b>29.21</b>	–	–	5	–	–	10
A*	<b>29.21</b>	<b>29.21</b>	0.0	–	–	–	9
MsAACO	<b>29.21</b>	<b>29.21</b>	0.0	9	9.80	0.72	7

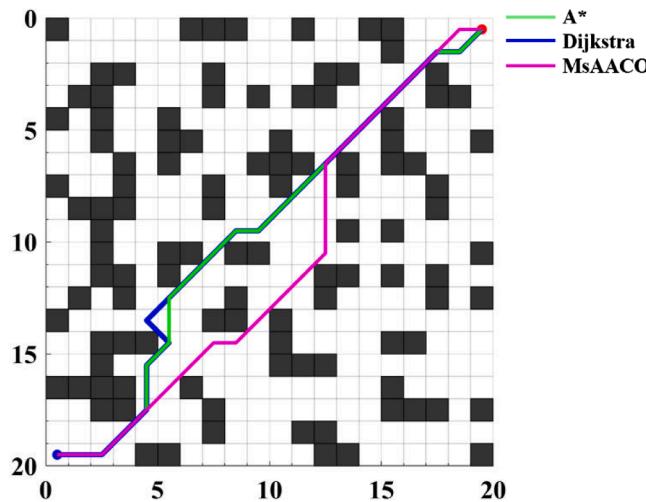


Fig. 26. Optimal paths obtained using MsAACO, Dijkstra's algorithm, and A\* algorithm.

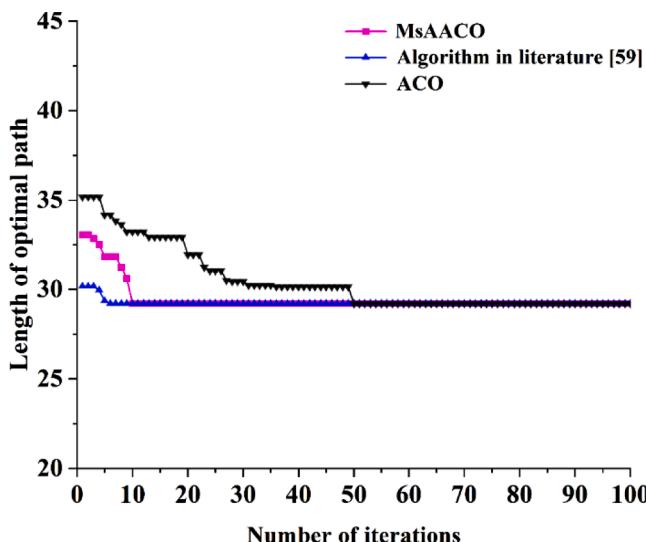


Fig. 27. Convergence curves of different algorithms.

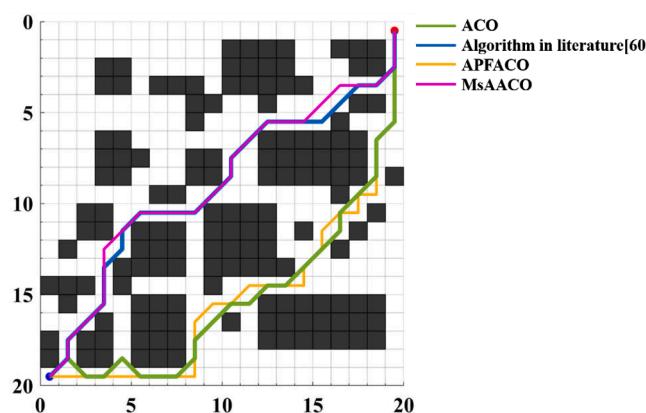


Fig. 28. Optimal paths generated by different ACOs.

algorithm in the space environment model of the Reference [60] to demonstrate the advantages of MsAACO. As shown in Fig. 28, the grid environment model size is  $20 \times 20$ , the starting node is the red dot, the

target node is the blue dot, and the coordinates of the starting and target nodes are  $(0.5, 19.5)$  and  $(19.5, 0.5)$ , respectively. In this study, the parameters of APFACO and the algorithm in Reference [60] were set consistently with those mentioned in Reference [60] to maintain their original performance. Therefore, the parameters setting in Reference [60] are:  $M = 50$ ,  $K = 100$ ,  $Q = 100$ ,  $\alpha = 1.1$ ,  $\beta = 6.0$ ,  $\rho = 0.2$ ,  $\lambda = 0.2$ ,  $N_0 = 5.0$ , and  $q_0 = 0.3$ . The common parameter setting of MsAACO is the same as that of the APFACO ( $M = 50$ ,  $K = 100$ ,  $\alpha = 1.1$ ,  $\beta = 6.0$ ,  $Q = 100$ , and  $\rho = 0.2$ ) to ensure the fairness of the experiment. All experiments were independently performed 20 times to avoid random errors. The statistical results of the algorithms are presented in Table 8. The generated optimal paths of these algorithms are displayed in Figs. 28–29, and the optimal convergence curves of the proposed MsAACO and other comparison algorithms are shown in Fig. 30.

From the experimental results in Table 8, the generated optimal path length of MsAACO, which is equal to the algorithm in Reference [60], and A\* algorithm (30.97), and less than that of Dijkstra's algorithm (32.14), APFACO (35.07), ACO (31.8), the MsAACO obtains the minimum “Best”, “Mean”, “Std.” of convergence generation ( $5.3 < 7.7 < 11.4 < 27.4$ ) and minimum turn times ( $13 < 14 < 15 = 15 < 19$ ) compared to the other four algorithms. Moreover, as shown in Figs. 28 and 29, the path-planning solution of MsAACO is shorter with fewer turn times compared to the other algorithms, which indicates that MsAACO has better stability and can effectively perform path planning in this experimental environment. It can be observed from Fig. 30 and Table 8 that the proposed MsAACO has an advantage in the convergence curve compared to the algorithms in the Reference [60] ( $5 < 6$ ). Moreover, Table 7 shows that the proposed MsAACO has the stability with the “Mean” of convergence iterations 5.3. Overall, the comprehensive environment results confirm that MsAACO is superior to other path-planning algorithms.

#### 4.6. Simulation experiment V

In this section, MsAACO is compared with six algorithms namely the best-first search, trace, original ACO, JPS, the algorithm in Reference [59], and the breadth-first search algorithm to verify its remarkable performance. To ensure fairness, the grid environment model of Reference [59] was utilized, as shown in Fig. 31. The grid environment model size was  $30 \times 30$  with coordinates  $(8.5, 0.5)$  for the starting node, represented by the blue dot, and the target node with coordinates  $(28.5, 25.5)$ , represented by the green dot. White grids represent free space and black grids represent obstacles. All the experiments were independently run 20 times to avoid random errors. The statistical results of MsAACO and the other path-planning algorithms are listed in Table 9. The optimal paths generated by these algorithms when solving the environmental models are shown in Figs. 31–32.

The experimental results in Table 7 show that the proposed MsAACO algorithm is superior to the best-first search algorithm, trace algorithm, original ACO, JPS algorithm, the algorithm in Reference [59], and breadth-first search algorithm. Through the “Best” of path length produced by MsAACO, which is equal to the breath-first-search algorithm

**Table 8**  
Experimental results of different algorithms.

Algorithms	Path length			Number of convergence generation			Turn times
	Best	Mean	Std.	Best	Mean	Std.	
ACO	31.80	33.76	1.05	17	27.4	3.34	19
APFACO	35.07	35.66	–	6	7.7	–	15
Dijkstra's	32.14	32.14	0.00	–	–	–	14
Algorithm in literature [60]	30.97	30.97	0.00	6	11.4	–	15
A*	30.97	30.97	0.00	–	–	–	15
MsAACO	30.97	30.97	0.00	5	5.3	0.20	13

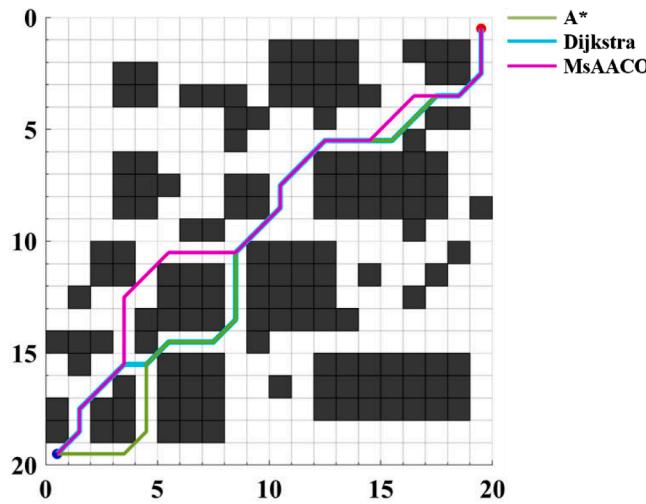


Fig. 29. Optimal paths obtained using MsAACO, Dijkstra's algorithm and A\* algorithm.

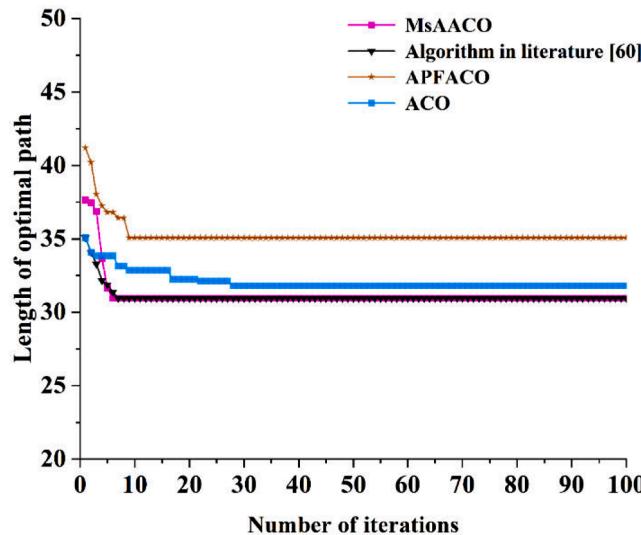


Fig. 30. Convergence curves of different ACOs.

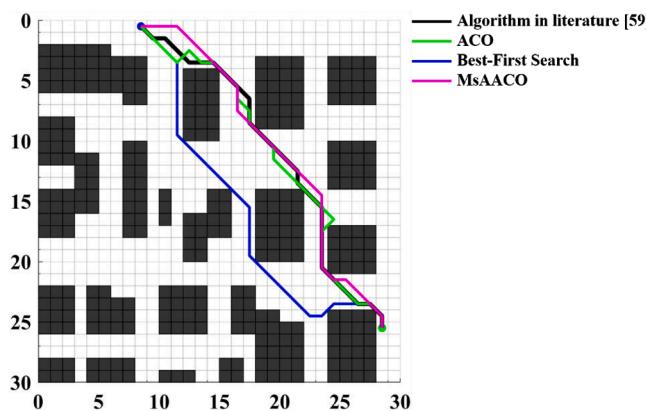


Fig. 31. Optimal paths generated by MsAACO, ACO, Best-First Search and the algorithm in Reference [59].

Table 9  
Experimental results of different algorithms.

Algorithm	Path length			Turn times
	Best	Mean	Std.	
Best-First Search	37.63	37.63	0.0	9
Trace	36.80	36.80	0.0	9
ACO	36.21	36.69	0.36	13
JPS	36.21	36.21	0.0	7
Algorithm in Reference [59]	35.63	–	–	13
Breadth-First-Search	35.63	35.63	0.0	8
MsAACO	35.63	35.63	0.0	8

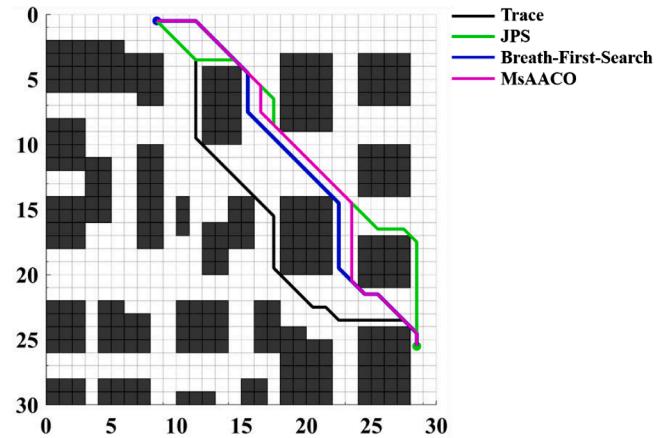


Fig. 32. Optimal paths generated by MsAACO, Trace, JPS, and Breadth-First Search.

and the algorithm in Reference [59] (35.63), and less than that of ACO (36.21), best-first search algorithm (37.63), JPS algorithm (36.21) and trace algorithm (36.80), the MsAACO can obtain lower turn times. Compared to JPS, MsAACO has a slight disadvantage in terms of turn times, whereas it performs better in terms of path length. From Figs. 31 and 32, benefiting from the proposed direction guidance mechanism, adaptive heuristic function, deterministic state transition probability rule, and non-uniform pheromone initialization, the generated optimal path of MsAACO is smoother and performs better than the other algorithms. Therefore, MsAACO is superior for path planning in complex environment models.

## 5. Conclusion

In this study, by combining the four proposed effective strategies, a multi-strategy adaptable ant colony optimization algorithm (MsAACO) was proposed to improve the deficiencies of ACO. First, a direction-guidance mechanism was proposed to make node selection directional. Then, the adaptive heuristic function introduced the location information, path length, and turn times of the starting and target nodes into the ACO. Moreover, the deterministic state transition probability rule made node selection faster. Finally, nonuniform pheromone initialization enhanced the ability of ACO to select advantageous regions. MsAACO was proposed by optimizing the major parameters of the strategies and combining the four strategies with ACO, and its performance was tested through effectiveness verification. Five environmental models and certain advanced algorithms, including variants of the ACO, A\* algorithm, Dijkstra's algorithm, JPS, best-first search, breadth-first search, trace, and excellent existing algorithms, were employed to validate the outstanding performance of MsAACO in addressing the path planning of mobile robots. Comprehensive statistical results demonstrate that MsAACO can achieve excellent path-planning solutions and has advantages in terms of path length, convergence speed, and turn

times. In addition, the obtained path-planning solutions of MsAACO were smoother and more stable than those of the other algorithms. Therefore, the proposed MsAACO can overcome the shortcomings of the original ACO and is better suited for mobile robot path-planning problems. In the future, MsAACO will also be applied to path planning for 3D environmental models and intelligent drilling [62,63].

### CRediT authorship contribution statement

**Junguo Cui:** Software, Methodology, Investigation, Writing – original draft, Data curation. **Lei Wu:** Supervision, Methodology, Funding acquisition. **Xiaodong Huang:** Software, Methodology, Formal analysis, Data curation. **Dengpan Xu:** Software, Writing – review & editing, Methodology, Conceptualization. **Chao Liu:** Writing – review & editing, Methodology, Formal analysis. **Wensheng Xiao:** Writing – review & editing, Supervision.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

Data will be made available on request.

### Acknowledgment

This research is funded by “the National Key R&D Program of China” (grant number 2021YFB3401400), the Major Scientific and Technological Innovation Project of Shandong Province, China (2022CXGC020405), the Taishan Scholars Program of Shandong Province (tsqn201909067), the Shandong Province Natural Science Foundation (ZR2020QE300), Fundamental Research Funds for the Central Universities (20CX06012A), and the Project of Ministry of Industry and Information Technology of the People’s Republic of China (Research on the key technology of treatment process for high -flow offshore natural gas, CJ09N20).

### References

- [1] T. Marcucci, M. Petersen, et al., Motion planning around obstacles with convex optimization, *Sci. Robot.* 8 (84) (2023) eadf7843.
- [2] Y.B. Niu, X.F. Yan, et al., An improved sand cat swarm optimization for moving target search by UAV, *Expert Syst. Appl.* 238 (2024) 122189.
- [3] S.W. Lin, A. Liu, et al., An improved fault-tolerant cultural-PSO with probability for multi-AGV path planning, *Expert Syst. Appl.* 237 (2024) 121510.
- [4] T. Zhao, Z.Y. Yan, et al., Hybrid path planning method based on skeleton contour partitioning for robotic additive manufacturing, *Robot. Comput. Integrat. Manuf.* 85 (2024) 102633.
- [5] C. Wang, X.T. Feng, et al., The optimization of an EV decommissioned battery recycling network: a third-party approach, *J. Environ. Manage.* 348 (2023) 119299.
- [6] C. Liu, L. Wu, et al., Improved multi-search strategy A\* algorithm to solve three-dimensional pipe routing design, *Expert Syst. Appl.* 122 (2024) 122313.
- [7] W.Y. Cai, S. Zhang, et al., Improved BINN-based underwater topography scanning coverage path planning for UAV in internet of underwater things, *IEEE Internet. Things J.* 10 (20) (2023) 18375–18386.
- [8] X. Li, S. Yu, Three-dimensional path planning for AUVs in ocean currents environment based on an improved compression factor particle swarm optimization algorithm, *Ocean Eng.* 280 (2023) 114610.
- [9] V.F. Yu, P.T. Anh, et al., A simulated annealing with variable neighborhood descent approach for the heterogeneous fleet vehicle routing problem with multiple forward/reverse cross-docks, *Expert Syst. Appl.* 237 (2024) 121631.
- [10] X. Liu, G. Li, et al., Agricultural UAV trajectory planning by incorporating multi-mechanism improved grey wolf optimization algorithm, *Expert Syst. Appl.* 233 (2023) 120946.
- [11] Y.Q. Chen, J.L. Guo, et al., Research on navigation of bidirectional A\* algorithm based on ant colony algorithm, *J. Supercomput.* 77 (2) (2021) 1958–1975.
- [12] G.X. Li, C. Liu, et al., A mixing algorithm of ACO and ABC for solving path planning of mobile robot, *Appl. Soft Comput.* 148 (2023) 110868.
- [13] B.B. Fan, L.L. Guo, An improved JPS algorithm for global path planning of the seabed mining vehicle, *Arab. J. Sci. Eng.* 15 (2023).
- [14] Srinivas Rakhee, M.B. cluster based energy efficient routing protocol using ANT colony optimization and breadth first search, *Procedia Comput. Sci.* 89 (2016) 124–133.
- [15] M.G. Zhang, B.J. Cheng, et al., A Fast algorithm of the shortest path ray tracing, *Chinese J. Geophys.* 49 (5) (2006) 1315–1323.
- [16] H. Huang, D.Q. Zhu, et al., Dynamic task assignment and path planning for multi-AUV system in variable ocean current environment, *J. Intell. Robot.* 74 (3–4) (2014) 999–1012.
- [17] X.B. Yu, W.G. Luo, Reinforcement learning-based multi-strategy cuckoo search algorithm for 3D UAV path planning, *Expert Syst. Appl.* 223 (2023) 119910.
- [18] X. Wu, H.L. Chen, et al., The autonomous navigation and obstacle avoidance for USVs with ANOA deep reinforcement learning method, *Knowl. Based Syst.* 196 (2020) 105201.
- [19] C. Huang, X.B. Zhou, et al., Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning, *Eng. Appl. Artif. Intell.* 121 (2023) 105942.
- [20] N. Ahmed, C.J. Pawase, et al., Distributed 3-D path planning for multi-UAVs with full area surveillance based on particle swarm optimization, *Appl. Sci.* 11 (8) (2021) 3417.
- [21] S.C. Xiao, X.J. Tan, et al., A simulated annealing algorithm and grid map-based UAV coverage path planning method for 3D reconstruction, *Electronics (Basel)* 10 (7) (2021) 853.
- [22] M.Y. Deng, Q.Q. Yang, Y. Peng, A real-time path planning method for urban low-altitude logistics UAVs, *Sensors* 23 (17) (2023) 7472.
- [23] X.B. Yu, N.J. Jiang, et al., A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning, *Expert Syst. Appl.* 215 (2023) 119327.
- [24] W. Zhang, S. Zhang, et al., Path planning of UAV based on improved adaptive grey wolf optimization algorithm, *IEEE Access* 9 (2021) 89400–89411.
- [25] W. Lee, G.H. Choi, T.W. Kim, Visibility graph-based path-planning algorithm with quadtree representation, *Appl. Ocean Res.* 117 (2021) 102887.
- [26] Z.Z. Wu, M.X. Huang, et al., Urban Traffic Planning and Traffic Flow Prediction based on ulchis gravity model and Dijkstra algorithm, *J. Physics: Conf. Series* 1972 (1) (2021) 012080–012090.
- [27] H.Q. Sang, Y.S. You, et al., The hybrid path planning algorithm based on improved A\* and artificial potential field for unmanned surface vehicle formations, *Ocean Eng.* 223 (2021) 108709.
- [28] F. Duchon, A. Babinec, et al., Path planning with modified a star algorithm for a mobile robot, *Procedia Eng.* 96 (2014) 59–69.
- [29] K.G. Akshay, A. Himanshu, et al., Time-efficient A\* algorithm for robot path planning, *Procedia Technol.* 23 (2016) 144–149.
- [30] C.K. Xiong, H.X. Zhou, et al., Rapidly-exploring adaptive sampling tree\*: a sample-based path-planning algorithm for unmanned marine vehicles information gathering in variable ocean environments, *Sensors* 20 (9) (2020) 2525.
- [31] P.C. Zhang, C. Xiong, et al., Path planning for mobile robot based on modified rapidly exploring random tree method and neural network, *Int. J. Adv. Robot. Syst.* 15 (3) (2018) 1729881418784221.
- [32] D.H. Kim, Y.S. Choi, et al., Adaptive rapidly-exploring random tree for efficient path planning of high-degree-of-freedom articulated robots, *Proc. Inst. Mech. Eng. part C-J. Mec. Eng. Sci.* 229 (18) (2015) 3361–3367.
- [33] Z. Sabir, S. Ben Said, et al., A reliable stochastic computational procedure to solve the mathematical robotic model, , *Expert Syst. Appl.* 238 (2024) 122224.
- [34] S.F. Xu, W.H. Bi, et al., A deep reinforcement learning approach incorporating genetic algorithm for missile path planning, *Int. J. Mach. Learn. Cybern.* 20 (2023).
- [35] L.L. Zhou, X.M. Ye, et al., An Improved genetic algorithm for the recovery system of usvs based on stern ramp considering the influence of currents, *Sensors* 23 (19) (2023) 8075.
- [36] Z.X. Jin, G.M. Luo, et al., WOA-AGA algorithm design for robot path planning, *Int. J. Comput. Commun. Control* 18 (5) (2023) 5518.
- [37] Y. Chong, H.Z. Chai, et al., Automatic recognition of geomagnetic suitability areas for path planning of autonomous underwater Vehicle, *Mar. Geodesy* 44 (7) (2021) 1–15.
- [38] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 53–66.
- [39] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, *IEEE transactions on systems, man, and cybernetics, Part B, Cybern. Publ. IEEE Syst., Man, Cybern. Soc.* 26 (1) (1996) 29–41.
- [40] B. Yang, L.W. Wu, et al., Location and path planning for urban emergency rescue by a hybrid clustering and ant colony algorithm approach, *Appl. Soft Comput.* 147 (2023) 110783.
- [41] C. Liu, L. Wu, et al., An improved heuristic mechanism ant colony optimization algorithm for solving path planning, *Knowl. Based Syst.* 271 (2023) 110540.
- [42] M.Z. Li, B. Li, et al., Optimized APF-ACO algorithm for ship collision avoidance and path planning, *J. Mar. Sci. Eng.* 11 (6) (2023) 1177.
- [43] H. Fatemidokht, M.K. Rafsanjani, F-Ant: an effective routing protocol for ant colony optimization based on fuzzy logic in vehicular ad hoc networks, *Neural Comput. Appl.* 29 (11) (2018) 1127–1137.
- [44] X.Y. Wang, L. Yang, et al., Robot path planning based on improved ant colony algorithm with potential field heuristic, *Control Decis.* 33 (10) (2018) 1775–1781.
- [45] C.W. Miao, G.Z. Chen, et al., Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm, *Comput. Indust. Eng.* 156 (2021) 107230.
- [46] X.X. Li, P. Hu, et al., Robot 3D path planning algorithm based on improved elitist potential field ant colony algorithm, *Comput. Sci. Appl.* 11 (4) (2021) 849–858.

- [47] Y. Tao, H. Gao, et al., A mobile service robot global path planning method based on ant colony optimization and fuzzy control, *Appl. Sci.* 11 (8) (2021) 3605.
- [48] J. Wei, J.J. Wang, et al., 3D path planning based on improved ant colony algorithm, *Comput. Eng. appl.* 56 (17) (2020) 217–223.
- [49] T. Hui, Research on robot optimal path planning method based on improved ant colony algorithm, *Int. J. Comput. Sci. Math.* 13 (1) (2020) 80.
- [50] N. Xiong, X.Z. Zhou, et al., Mobile robot path planning based on time taboo ant colony optimization in dynamic environment, *Front. Neurorobot.* 15 (2021) 642733.
- [51] Z. Pei, X. Chen, Improved ant colony algorithm and its application in obstacle avoidance for robot, *CAAI Trans. Intell. Syst.* 10 (1) (2015) 90–96.
- [52] J.L. Matez-Bandera, J. Monroy, et al., Efficient semantic place categorization by a robot through active line-of-sight selection, *Knowl. Based Syst.* (2021) 108022.
- [53] S.F. Feng, Q. Lei, et al., Mobile robot path planning based on adaptive ant colony algorithm, *Comput. Eng. Appl.* 55 (17) (2019) 35–43.
- [54] C.W. Miao, G.Z. Chen, et al., Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm, *Comput. Ind. Eng.* 156 (2021) 107230, 2021.
- [55] L. Wu, X. Tian, et al., An improved heuristic algorithm for 2D rectangle packing area minimization problems with central rectangles, *Eng. Appl. Artif. Intell.* 66 (2017) 1–16.
- [56] B. Shuang, J.P. Chen, Z.B. Li, Study on hybrid PS-ACO algorithm, *Appl. Intell.* 34 (1) (2011) 64–73.
- [57] S.C. Zhang, J.X. Pu, et al., An adaptive improved ant colony system based on population information entropy for path planning of mobile robot, *IEEE Access* 9 (2021) 24933–24945.
- [58] D.H. Zhang, X.M. You, et al., Dynamic multi-role adaptive collaborative ant colony optimization for robot path planning, *IEEE Access* 8 (2021) 129958–129974.
- [59] J. Zhao, D.D. Cheng, et al., An improved ant colony algorithm for solving the path planning problem of the omnidirectional mobile vehicle, *Math. Probl. Eng.* (2016) 7672839, 2016.
- [60] Q. Luo, H.B. Wang, et al., Research on path planning of mobile robot based on improved ant colony algorithm, *Neural Comput. Appl.* 32 (6) (2020) 1555–1566.
- [61] J.H. Liu, J.G. Yang, et al., Robot global path planning based on ant colony optimization with artificial potential field, *Trans. Chinese Soc. Agricul. Mach.* 46 (9) (2015) 18–27.
- [62] X. Yuqiang, W. Yucong, et al., Experimental study on ultrasonic wave propagation characteristics of gas-liquid two-phase flow in riser annulus, *Appl. Ocean Res.* 141 (2023) 103771.
- [63] Y. Xu, F. Li, et al., Numerical simulation method and structural optimization for shearing capacity of ram blowout preventers, *Geoenergy Sci. Eng.* (2023) 212559.