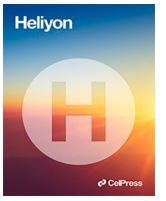
Heliyon 10 (2024) e27753

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

Improved ant colony optimization for safe path planning of AUV

Meng Ronghua [a,b,1], Cheng Xinhao [a,b,1], Wu Zhengjia [a,b,*], Du xuan [a,b]

[a] Hubei Key Laboratory of Hydroelectric Machinery Design and Maintenance, China Three Gorges University, Yichang, Hubei, 443002, China
[b] Intelligent Manufacturing Innovation Technology Center, China Three Gorges University, Yichang, Hubei, 443002, China

ARTICLE INFO

Keywords:
Improved ant colony optimization
Safety factors
Dam inspections

ABSTRACT

In order to address the autonomous underwater vehicle navigation challenge for dam inspections, with the goal of enabling safe inspections and reliable obstacle avoidance, an improved smooth Ant Colony Optimization algorithm is proposed for path planning. This improved algorithm would optimize the smoothness of the path besides the robustness, avoidance of local optima, and fast computation speed. To achieve the goal of reducing turning time and improving the directional effect of path selection, a corner-turning heuristic function is introduced. Experimental simulation results show that the improved algorithm performs best than other algorithms in terms of path smoothness and iteration stability in path planning.

1. Introduction

Chinese dams were mostly built between the 1950s and 1980s [1], and due to low design standards, poor construction quality, inadequate management systems, and insufficient dam risk research [2], some small dams from that period experienced premature aging problems. Their actual safety lifespan was even less than 30 years [3], which not only affected project efficiency but also posed serious safety risks [4].

Therefore, regular inspections of dams are necessary for their normal and safe operation. Path planning is an important field of research in autonomous underwater vehicle (AUV) [5]. Path planning can be categorized based on the environment into static and dynamic scenarios [6]. In a static environment, both the starting point and the destination are fixed [7], and obstacles are also stationary. In contrast, dynamic environments involve obstacles that are constantly in motion [8]. When planning the path for a robot, environmental recognition can be classified into global path planning and local path planning [9]. Global path planning represents a scenario where obstacles are known before planning [10]. Local path planning refers to an unknown environment where suitable paths are planned by scanning the environment map with radar to locate obstacles [11]. The methods used in path planning include the artificial potential field method [12], particle swarm optimization [13], A-Star algorithm [14], ant colony algorithm [15], genetic regulatory networks [16], genetic algorithm [17], D-star algorithm [18], and neural network algorithm [19]. The ant colony algorithm (ACO) is based on the foraging process of ants in nature. This algorithm continuously updates the current path until finding an optimal path. Due to its strong robustness and good information feedback, the ant colony algorithm has gained widespread attention. However, it has issues such as slow convergence, low efficiency, and susceptibility to local optima.

In response to the aforementioned issues, Pei and Chen [20]proposed a dynamic adaptive parameter adjustment strategy for the

* Corresponding author.Hubei Key Laboratory of Hydroelectric Machinery Design and Maintenance, China Three Gorges University, Yichang, Hubei, 443002, China.
E-mail address: zjwu@ctgu.edu.cn (W. Zhengjia).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.heliyon.2024.e27753
Received 19 September 2023; Received in revised form 27 January 2024; Accepted 6 March 2024
Available online 20 March 2024

ACO method by integrating a generalized pheromone updating rule. This strategy effectively improves the search efficiency and solution quality for obstacle avoidance in mobile robots. Liu, Yang, Liu, Tian, and Gao [21] introduced a new improved ACO algorithm that combines pheromone diffusion and geometric local optimization, resulting in faster convergence of ACO. Wang, Yang, Zhang, and Meng [22] utilized the artificial potential field method to adjust the initial value of ACO pheromone, set an iteration threshold, and adjust the evaporation coefficient to prevent ACO from getting trapped in local optima and enhance its search capability. Zhang, Wang, Fu, and Su [23] used random value initialization for pheromone and performed crossover operations to improve the search capability of ACO, although at the cost of reduced convergence speed. Sui F [24] proposed a dual-layer hybrid algorithm called ACO + PSO + A* for multi-task path planning in Autonomous Underwater Vehicles (AUVs). In this algorithm, the concept of adaptive pheromone evaporation rate is introduced, along with parameter settings and algorithm performance evaluation. The improvements in the algorithm mainly focus on path planning and obstacle avoidance strategies to address complex marine environments. Wu L [25] introduced enhanced heuristic functions, adaptive pseudo-random transfer rules, and a method for unevenly distributed pheromones. This approach, combined with the traditional ACO algorithm, resulted in an improved adaptive ACO algorithm. Ntakolia C [26] proposed an enhanced ant colony algorithm for local path planning under multi-modal constraint conditions to improve the path planning efficiency of Unmanned Surface Vehicles in response to complex marine environments. Xie X [27] combined ant colony algorithm and chaos optimization algorithm, introducing information pheromone differential update strategy and local search optimization strategy. Through the chaos optimization algorithm, a large number of paths are randomly generated, and pheromones are left on paths with lower effective doses. The information pheromone differential update strategy updates the pheromone concentration based on the effective dose of the path, increasing the attractiveness of paths with lower radiation doses, thus improving the algorithm's efficiency. Additionally, a local search optimization algorithm is introduced to find paths with lower effective doses. He P [28] proposed a self-learning ant colony optimization algorithm by combining ant colony algorithm with a self-learning mechanism to address path planning problems in dynamic public transportation networks.

However, most existing path planning methods rarely consider path smoothness and safety. In the case of AUV underwater inspections, neglecting path smoothness can result in excessive energy consumption due to underwater resistance, leading to premature termination of inspection tasks due to battery depletion. It can also cause instability and affect the detection devices on the AUV, thereby impacting the overall inspection results. Hence, an improved Smooth Ant Colony Optimization (SACO) is proposed, with the objective of achieving the shortest path length and highest smoothness. This algorithm solves the path nodes for AUV in a known working environment, determining the sequence of path nodes for the AUV.

This article's contributions are as follows.

- Conducting two rounds of expansion on irregular obstacles that do not fully occupy the grid to prevent collisions and ensure safety.
- To address the issue of ant colony algorithms easily falling into local optima, a new pheromone update strategy has been designed.
- Designing a fitness function that comprehensively considers path length and global tortuosity. The objective is to achieve optimized overall performance by minimizing planned path length and reducing abrupt changes.
- Through a series of simulations and experiments, our method has been demonstrated to outperform existing works in terms of path length, path smoothness, computational speed, and other aspects.

2. AUV path planning problem and mathematical model

a.a Problem Description

The AUV path planning problem studied in this paper belongs to the category of priori-based global path planning [29], which can be described as follows: An AUV is tasked with conducting survey operations at known locations within a designated work area. From the above description, it is evident that this is an AUV path planning problem, with the primary objective of finding a path between the AUV's starting and ending positions, ensuring collision-free navigation, and minimizing the number of path segments and turning angles during the AUV's trajectory.

a.b Establishment of AUV Mathematical Model

During AUV dam inspection missions, a lack of path smoothness in the planned trajectory can lead to abrupt changes and sharp turns, which hinder accurate identification. Conversely, optimizing path length not only enhances economic benefits and cost-saving but also improves task execution efficiency. Consequently, to holistically address various aspects of dam inspection tasks, a mathematical model is proposed. The objective is to minimize both the planned path length and the occurrence of abrupt changes, aiming to achieve an optimized comprehensive performance.

$$f_1 = \min\{PL\} \tag{1}$$

$$f_2 = \min\{P\theta\} \tag{2}$$

$$s.t. \quad \forall(x,y) \notin R_f \quad f(x,y) = 0 \tag{3}$$

$$L_t < L_{\max} \tag{4}$$

$$\theta_i \leq \theta_{max} \tag{5}$$

The main constraints in AUV path planning include.

(1) Terrain constraint: To ensure the safety of the AUV during navigation, it is necessary to avoid areas with terrain obstacles, which are defined as regions that the AUV cannot traverse underwater. Let the set of non-traversable regions in the planning area be denoted as $R_f^{'}$. To more effectively consider the actual size of robots and factors such as positioning errors that may exist during path execution, it is necessary to subject obstacles to a certain degree of inflation processing. This ensures that the generated path will not lead to collisions during actual execution, thereby enhancing the safety and feasibility of the path. Expanding the range of the original obstacles yields $R_f = \varepsilon R_f^{'}$. The inflation factor ε is determined based on the actual size of the AUV. As a result, the terrain constraint in the planning area can be represented by the path curve $f(x,y) = 0$, where (x,y) represents the latitude and longitude coordinates of any point on the path and satisfies the constraint in Equation (3).
(2) Maximum range constraint: The farthest distance that the AUV can travel is limited by its onboard energy. Therefore, the energy constraint needs to be considered in the path planning. The AUV's range constraint can be represented by Equation (4), where $L_{\max}$ represents the maximum distance that the AUV can travel and L_t represents the path length planned by the algorithm.
(3) Turning angle constraint: Due to the limitations of its maneuvering capabilities, the AUV needs to adhere to a certain range of angles when making turns. The maximum allowable turn angle of the AUV is θ_{max}, and for any turning angle θ_i, the constraint is represented by Equation (5).

The formulas for calculating the path length and path tortuosity in the objective function are as follows:

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{6}$$

$$PL_t = \sum_{i=1,j=i+1}^{i=n-1,j=n} d_{ij} \tag{7}$$

$$\theta_i = \cos^{-1}\frac{(x_i - x_s) \times \left(x_j - x_i\right) + (y_i - y_s) \times \left(y_j - y_i\right)}{\sqrt{\left(x_i - x_s\right)^2 + \left(y_i - y_s\right)^2} \times \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2}} \tag{8}$$

$$P\theta_t = \sum_{i=2}^{i=n-1} \theta_i \tag{9}$$

Formulas (6) and (7) are used to calculate the path length PL_t of the t-th path. Formulas (8) and (9) are used to calculate the local path tortuosity $P\theta_t$ for the t-th path.

Formula (8) calculates the local path tortuosity θ_{ij}^t for the i-th search of the t-th path. θ_{ij}^t represents the angle formed between the vectors $\overrightarrow{si}$ (formed by the $i-1$-th search result s and the i-th search result i) and $\overrightarrow{ij}$ (formed by the i-th search result i and the $i+1$-th search result j). $\theta_i \in [0°, 45°, 90°, 135°]$ It indicates the amount of direction change when the AUV follows the planned path. A smaller value indicates a smoother path.

Formula (9) calculates the path tortuosity θ_{ij}^t of a given path by cumulatively adding up the values of pt formed by each set of three consecutive path points until reaching the $i-1$-th path point. The resulting value represents the degree of curvature in the path. A smaller value indicates a smoother and less tortuous path, which is generally considered better.

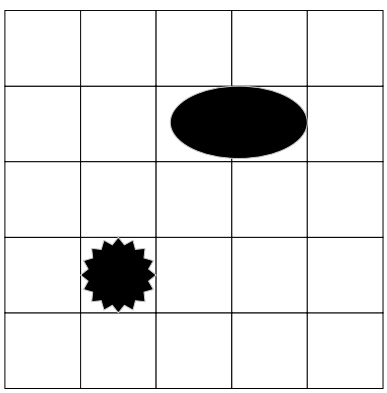


Fig. 1. Actual obstacles.

3. Environment modeling

In practical applications, AUV often operate in complex underwater environments filled with obstacles or pipelines. To accurately plan the path for the robot, a two-dimensional modeling of the AUV's working environment is required. The grid-based method is used for environmental modeling, where the maximum side length of the AUV is taken as the unit grid length "l" to facilitate the robot's turning direction.

The areas containing the lathe and the shelf are designated as obstacle regions (see Fig. 1).If there are irregular obstacles that do not completely occupy a grid, they are processed for initial inflation to fully occupy the grid (see Fig. 2).Based on this, further inflation is performed on the obstacles to prevent collisions, with the inflation size set to l/2, and the grid activity value is set as 1 for the inflated areas(see Fig. 3).

The remaining areas are designated as free movement regions with a grid activity value of 0, indicating that the AUV can freely navigate through these areas without any obstacles.

The grid index is equal to the corresponding index of the grid in the map matrix. The formula for converting the index to the grid is as follows:

$$i_i = mod\left(\frac{i - \left|\frac{i}{m\times n}\right| \times m \times n - 1}{2}\right) + 0.5 \tag{10}$$

$$ji = fix\left(\frac{i - \left|\frac{i}{m\times n}\right| \times m \times n - 1}{m}\right) + 0.5 \tag{11}$$

The AUV is abstracted as a particle, and its movement rules are as follows:

The AUV can freely move in the free grids, but it cannot move in the black obstacle regions. However, since the obstacles have undergone inflation processing, the AUV can pass through the corners of the obstacle grids.

During the operation of the AUV, a taboo list is set up to prevent it from passing through the same point repeatedly.

In each unit of time, the robot can move one grid in one of the eight directions: up, down, left, right, as well as upper right, lower right, upper left, and lower left.

4. Algorithm design

c.a Ant Colony Algorithm

The ant algorithm is a biomimetic algorithm derived from the natural world of biological organisms. As a general-purpose stochastic optimization method, it incorporates the behavioral characteristics of ants in the insect kingdom and has been effective in solving a series of difficult combinatorial optimization problems through its inherent search mechanism.

Based on observations and research by entomologists, it has been found that ants in the biological world have the ability to find the shortest path from their nest to a food source without any visible cues. They can adaptively search for new paths and make new choices as the environment changes. When ants, as insects, search for food sources, they can release a unique secretion called pheromone on the path they have traveled. This allows other ants within a certain range to detect and be influenced by it, affecting their future behavior. As more ants pass through certain paths, they leave behind more pheromone trails, which leads to an increase in the intensity of the pheromone (pheromone gradually diminishes over time). Consequently, later ants have a higher probability of choosing that path, further increasing the intensity of the pheromone on that path. Since the principle is a positive feedback mechanism, the ant

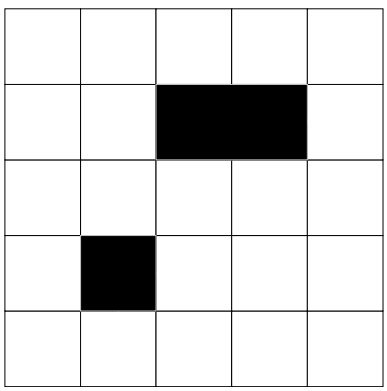


Fig. 2. First expansion.

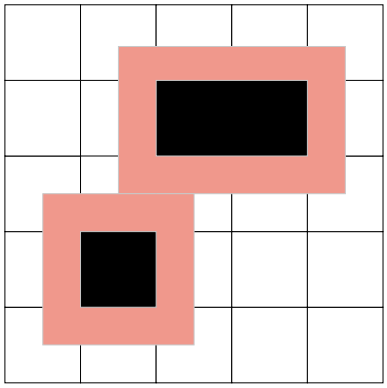


Fig. 3. Secondary expansion.

algorithm can be understood as an enhanced learning system. The basic principle is as follows:

When ants move, they select paths based on the concentration of pheromone and a distance heuristic function. The probability of an ant moving from one grid to another can be expressed as:

$$P_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s \in A_k} \tau_{iu}^{\alpha}(t)\eta_{iu}^{\beta}(t)} & s \in A_k \\ 0 & s \notin A_k \end{cases} \tag{12}$$

In this context, α represents the pheromone heuristic factor, and β represents the distance expectation function factor. They respectively influence the importance of pheromones and distance in the heuristic functions. A_K denotes the set of possible destinations for the ant to reach in the next step. $\tau_{ij}(t)$ represents the concentration of pheromones on the current path at time t. η_{ij}(t) represents the distance heuristic function.

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \tag{13}$$

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{14}$$

Each ant leaves behind some pheromones while moving, so as the algorithm iterates, the pheromones accumulate and also gradually evaporate. After all ants complete their first iteration, the pheromone concentration on the paths is refreshed according to the following equation:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{15}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{16}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} \\ 0 \end{cases} \tag{17}$$

In this context, ρ represents the pheromone evaporation coefficient; k represents the number of ants. $\Delta\tau_{ij}$ represents the sum of added pheromones between two points; $\Delta\tau_{ij}^{k}$ represents the increment of pheromones on the two points; L_k represents the length of the path traveled by ants, and Q represents the pheromone reinforcement coefficient.

c.b Improvement of Ant Colony Algorithm

If only basic ant colony algorithms are used for robot path planning, the consideration of path factors in the heuristic function is limited to local paths using greedy algorithm thinking. This approach not only tends to produce local optima but is also not the best choice when analyzing from a global path perspective. Furthermore, neglecting the influence of local paths may affect the precision of machine detection during inspections and even lead to safety concerns. In response to the above issues, reasonable optimization strategies and improvement solutions are proposed which include introducing a corner heuristic function to enhance path selection direction and reduce turning time. Additionally, improving the heuristic function enables the rapid identification of optimal paths.

4.1. Penalty factor

To penalize solutions that do not meet the objective and make the algorithm more inclined to search for satisfactory solutions, a penalty factor is introduced.

$$P_t^{'} = \begin{cases} 1, P_t(end) = goalpoint \\ 0, else \end{cases} \tag{18}$$

$$P_t = \left(1 - P_t^{'}\right) \times 10000 \tag{19}$$

Formula (17) represents whether the search path reaches the target point. If it does $P_t^{'} = 1$. When the constraint conditions for a successful search path are not met $P_t^{'} = 0$.

4.2. Improvement in pheromone update strategy

In order to address the issue of the ant colony algorithm getting stuck in local optima, this paper has made improvements to the pheromone update strategy. The specific modifications are as follows:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\left(\Delta\tau_{ij} + \tau^*\right) \\ \Delta\tau_{ij} = \sum_{k=1}^{n} \Delta\tau_{ij}^k \end{cases} \tag{20}$$

$$\tau^* = \frac{\tau_{max} - \Delta\tau_{ij}}{\tau_{ij}(t)} \tag{21}$$

In this context, where τ^* is the pheromone adjustment factor, and τ_{max} is the maximum pheromone value. When the pheromone level $\tau_{ij}(t)$ at a certain point is excessively high, the increment of the pheromone, represented as $\Delta\tau_{ij}$, will also increase. Consequently, as $(\tau_{max} - \Delta\tau_{ij})$ decreases, τ^* will decrease as well. Therefore, τ^* effectively adjusts the pheromones at that location for the next time step, preventing the pheromones at various points from getting disproportionately larger and thus avoiding local optima in the pheromone distribution.

4.3. Designing a holistic fitness function

In order to ensure the effectiveness of comparing the objectives, given that the path length PL_t and global curvature $P\theta_t$ vary on different scales, we will use the Min-Max normalization method to normalize both indicators before designing the holistic fitness function.

Normalization formula:

$$K\left(x^k\right) = \frac{x^k - x_{min}}{x_{max} - x_{min}} \tag{22}$$

The numerical range of path curvature and path length in the search results is dependent on the spatial position of the starting point (x_s, y_s) and the target point (x_E, y_E).

$$fit_t = P_t + \alpha k(PL_t) + \beta k(P\theta_t) \tag{23}$$

The weights "α" and "β" represent the respective factors. Based on the relative importance of the two optimization objectives in the practical application of AUV, it can be deduced from the definition of the comprehensive fitness function mentioned above that the smaller the corresponding fitness value, the higher the superiority of the path.

4.4. Safety heuristic function

Taking into consideration that excessive turns during the robot's movement will not only increase the walking distance but also pose a higher level of danger, this paper introduces a safety heuristic function φ to enhance the stability of the robot during its traversal. This is done in order to mitigate the risks associated with a high number of turns.

$$\varphi_{ij}(t) = \begin{cases} \zeta r & dir_{ij}(t) = dir_{iv}(t) \\ \dfrac{(1-\zeta)r}{L_J} & otherwise \end{cases} \tag{24}$$

In the equation: r represents the heuristic constant; ζ denotes the safety degree constant; L_J is the size of the set of unvisited grid cells; $dir_{ij}(t)$ represents the transition direction from current grid node i to the next grid node j at moment t; and $dir_{iv}(t)$ represents the

transition direction from previous grid node v to current grid node i at moment t. From this formula, it can be inferred that when the current transfer direction $dir_{ij}(t)$ at time t is consistent with the previous transfer direction $dir_{iv}(t)$, it will increase the likelihood of moving in the same direction. This encourages the robot to move along a straight line as much as possible

4.5. Improved heuristic function

In the basic ant colony algorithm, the heuristic function relies solely on the distance d_{ij} between the current grid cell i and the candidate grid cell j. However, using this method alone does not establish a connection with the destination, resulting in a weak routing effect of the distance heuristic function. The ants are unable to effectively utilize both the pheromone concentration and the distance between the grid cells. This diminishes the search efficiency and the ultimate pathfinding result of the ant colony algorithm. Therefore, this paper proposes an improved heuristic function, where the value of the function is determined by both the distance d_{ij} between the current grid cell i and the candidate grid cell j, and the distance d_{je} between the candidate grid cell j and the target grid cell e, as shown in Equation (22):

$$\eta_{ij}(t) = \frac{1}{\varpi \times d_{ij} + \lambda \times d_{je}} \tag{25}$$

Where d_{je} represents the distance between the candidate grid cell j and the target grid cell e, d_{ij} represents the distance between the current grid cell i and the candidate grid cell j, d_{se} represents the distance between the starting grid cell s and the target grid cell e. The values of ϖ and λ represent amplification factors that can be adjusted based on the environment.

To facilitate the computation of the heuristic function, a matrix of adjacent grid cell distances $D_{n^2\times 8}$ is established for each grid cell as shown in Equation (23).

$$D(i,j) = \begin{cases} l, G(i) = 0 \& \mathrm{mod}(b,2) = 1 \& G(j) = 0 \& G(i_1) + G(i_2) + G(i_3) \neq 3 \\ \sqrt{2} \times l, G(i) = 0 \& \mathrm{mod}(b,2) = 0 \& G(j) = 0 \& G(i^{'}) + G(i^{''}) = 0 \& G(i_4) + G(i_5) \neq 2 \\ \infty, else \end{cases} \tag{26}$$

The equation contains several symbols: l represents the width of a single grid cell, $G(i)$ represents the state of grid cell i, with a value of 0 indicating a feasible grid cell and 1 indicating the presence of an obstacle. j represents the state of the candidate grid cell, with a value of 0 indicating a feasible grid cell and 1 indicating the presence of an obstacle. b represents the directional index of the candidate grid cell relative to i, with its value determined as shown in Figure (4). i_1, i_2, i_3 represents the three neighboring grid cells adjacent to the candidate grid cell, while $i_4, i_5, i^{'}, i^{''}$ represents the neighboring grid cell perpendicular to the diagonal direction. The specific arrangement can be seen in Figure (4).

Path local turning degree heuristic function:

$$\mu = e^{1-\frac{\theta_{ij}}{\pi}} \tag{27}$$

The symbol θ_{ij} represents the angle between the line connecting the starting grid s and the current grid i, and the line connecting the current grid i and the candidate grid j.

The formula for the path selection probability is as follows:

$$P^{m}{}_{ij}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \times [\eta_{ij}(t)]^{\beta} \times [\mu_{ij}(t)]^{\gamma} \times [\varphi_{ij}(t)]^{\varepsilon}}{\sum_{u\in allowed_u} [\tau_{iu}(t)]^{\alpha} \times [\eta_{iu}(t)]^{\beta} \times [\mu_{ij}(t)]^{\gamma} \times [\varphi_{ij}(t)]^{\varepsilon}} & if\ j \in allowed_u \\ 0 \quad other \end{cases} \tag{28}$$

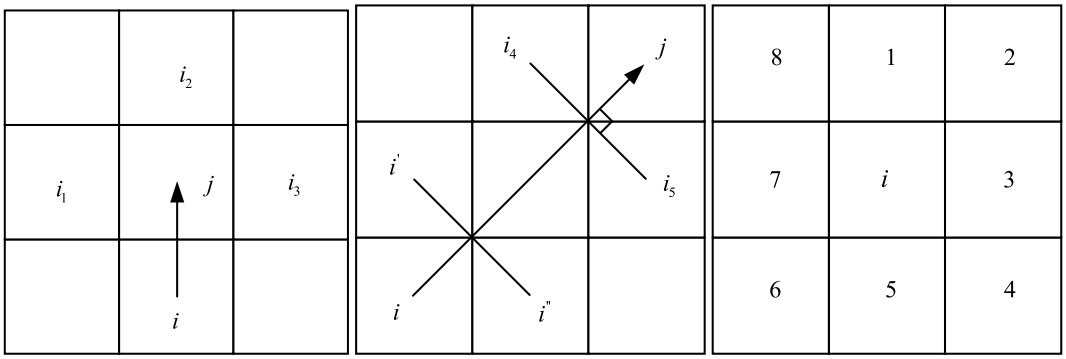


Fig. 4. Grid distance matrix.

c.c Solving Process

The algorithm procedure is as follows.

(1) Initialization: Set the number of iterations, population size, and number of ants. Randomly generate an initial path.
(2) Ant movement: Following the rules of the ant colony algorithm, each ant selects the next node to move to based on a certain strategy. During the movement, ants deposit pheromones.
(3) Pheromone update: There are two ways to update the pheromone strength based on feedback information: global update and local update.
(4) Best solution selection: After each iteration, evaluate the path length and smoothness of each ant and select the best path for that round.
(5) Smoothness optimization: Apply a path smoothing algorithm to optimize the best path, improving its overall smoothness.
(6) Termination criterion: Continue the iteration process until the specified number of iterations is reached or the iteration results are consistently the same for multiple generations. Once the stopping criteria are met, indicating the lowest energy consumption path has been found, further smoothing is applied to obtain a smooth path suitable for AUV tracking.

5. Experiment and simulation

d.a Software Development Environment

The MATLAB R2023a simulation software was used in this study to simulate the traditional ant colony algorithm and the improved algorithm. The simulation is focused on the path planning process of a mobile robot on a ground with obstacles. Since the simulation process of the ant colony algorithm involves roulette wheel selection, each run may produce different results. Therefore, multiple simulations were conducted for each algorithm in the experiment, and the central results were taken to evaluate the superiority or inferiority of the algorithms.

d.b Orthogonal Test

In the experiment, following empirical practices, the ant colony algorithm parameters are initialized as: $M = 40$, $T = 60$, $\alpha = 2$, $\beta = 7$, $Q = 1$, $\rho = 0.5$. On this basis, in order to test the impact of the introduced safety factor and tortuosity factor on algorithm performance, orthogonal experiments are conducted on these two parameters. According to experience, the safety factor is set with three values: 1, 2, and 3, while the curvature factor is set with three values: 2, 3, and 4. Under the same conditions(see Fig. 5), with a fixed number of 60 iterations and other parameters held constant, a total of 9 combinations are tested. Each combination is tested four times, resulting in a total of 36 tests for the number of turns, path length, and average fitness. The results, as presented in Table 1, are sorted based on fitness.

From the statistical analysis, it is evident that the fitness is optimal when parameter $\varepsilon = 2, \gamma = 3$ is introduced. Therefore, this study incorporates the parameter $\varepsilon = 2, \gamma = 3$ for experimentation.

d.c Algorithm Simulation and Comparative Experiment

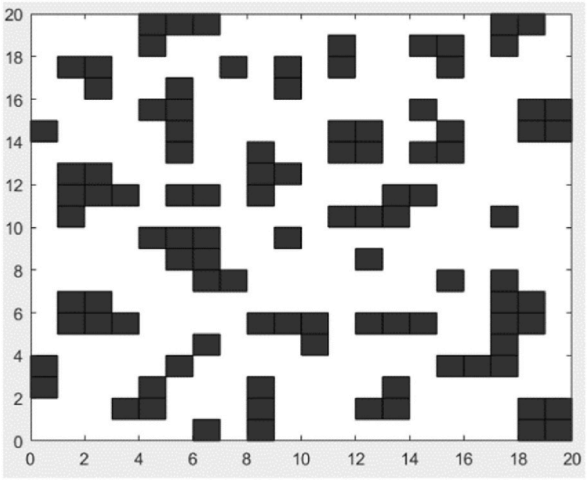


Fig. 5. 20 × 20 environmental map.

Table 1
Orthogonal experiment.

ε	γ	PL	$P\theta$	fitness
3	4	33.7	247.5	0.8158
2	4	34.7	202.5	0.7719
1	2	33.4	234	0.7289
1	4	35.2	166.5	0.7000
3	3	33.7	193.5	0.6158
1	3	33.7	180	0.5658
2	2	32.8	166.5	0.4053
3	2	30.6	207	0.2851
2	3	29.5	171	0.0167

To validate the SACO algorithm's ability to avoid local optima and exhibit good robustness in robot path planning, ACO and IACO [30]are chosen as comparative algorithms. Comparative simulation experiments are conducted on three different environmental maps. The parameters for the three algorithms in a 20 × 20 map are presented in Table 2.

5.1. 20 × 20 simulation environment

In the simulation experiment conducted on a 20 × 20 map, the ACO algorithm, IACO algorithm, and the newly proposed SACO algorithm are utilized for robot path planning. The convergence curve graphs and path planning diagrams for these algorithms are shown in Figs. 6 and 7 respectively. The experimental results are presented in Table 3 and Table 4.

From the analysis based on Fig. 7, it can be observed that the ACO algorithm has the slowest convergence rate and has not fully converged by the end, resulting in the longest optimal path among the three algorithms. The IACO algorithm requires more iterations to achieve convergence compared to the other algorithms. In contrast, the proposed SACO algorithm yields the shortest path. As shown in Table 3, the path length obtained by SACO algorithm is reduced by 12.52% compared to ACO algorithm and by 3.92% compared to IACO algorithm. Moreover, there is also a significant improvement in computational speed. Additionally, the number of inflection points in SACO algorithm is reduced by 16.66% compared to ACO algorithm and by 50% compared to IACO algorithm.

5.2. 30 × 30 simulation environment

To validate the robot movement path planning capabilities of the proposed SACO algorithm under different scenarios, simulations of robot path planning were conducted on a 30 × 30 grid map using ACO, IACO, and SACO algorithms. Asthe terrain is more complex compared to a 20 × 20 map, to maintain algorithm reliability, the number of ants was set to 100, the maximum iteration count was set to 150, and other parameters remained unchanged. The path index is shown in Table 5.The path planning results and algorithm convergence curves for the three algorithms are shown in Figs. 8 and 9, respectively.

From Fig. 9 and Table 6, it can be observed that the ACO algorithm has a slow search speed, a longer search for the optimal path, and is difficult to obtain the optimal solution. The IACO algorithm exhibits some blindness in the early stages of search and converges slowly. In contrast, the proposed SACO algorithm converges quickly in the early stages of search and finds the shortest path. According to Table 6, the shortest path obtained using the SACO algorithm is 12.6% shorter than ACO and 2.5% shorter than IACO. The system's runtime is reduced by 53.3% and 35.7%, respectively, compared to ACO and IACO. The number of turning points is reduced by 29.4% compared to ACO. Therefore, the SACO algorithm has higher search efficiency, fewer turning points, smoother paths, and strong path planning performance.

5.3. 40 × 40 simulation environment

To verify the robot path planning capabilities of the SACO algorithm in scenarios with a larger environmental space and more complex obstacle distribution, a 40 × 40 grid map was created. Robot path planning simulations were conducted using ACO, IACO, and SACO algorithms. Due to the larger size and more complex obstacle distribution of the 40 × 40 map, to maintain algorithm reliability and stability, the number of ants was set to 100, the maximum iteration count was set to 400, and other parameters remained unchanged. The path index is shown in Table 7 .The path planning results and algorithm convergence curves for the three algorithms are shown in Figs. 10 and 11, respectively

From Figs. 10 and 11, it can be observed that when the robot's working environment is large, and the distribution of obstacles is

Table 2
Algorithm parameters setting.

	M	T	α	β	Q	ρ	q_o	ε	γ	$\omega 1$	$\omega 2$
ACO	40	60	2	7	1	0.5	–	–	–	–	–
IACO	40	60	2	7	1	0.5	$0.1+2\cdot(k-0.45T)^2/T^2$	–	–	1	0
SACO	40	60	2	7	1	0.5	–	2	3	–	–

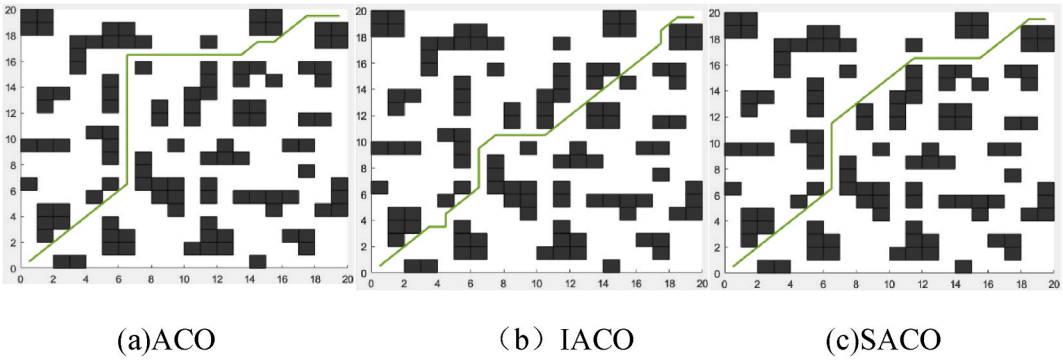


Fig. 6. Optimal planning path based on three ant colony algorithms.

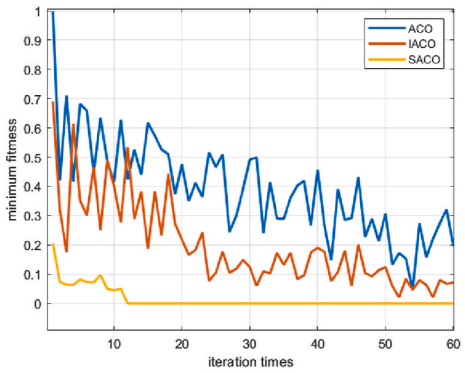


Fig. 7. Fitness iteration diagram.

Table 3
Optimal path result.

	Optimal Path Result
ACO	1 22 43 64 85 106 127 147 167 187 207 227 247 267 287 307 327 328 329 330 331 332 333 334 355 356 377 398 399 400
IACO	1 22 43 64 65 85 106 127 147 167 187 208 209 210 211 232 253 274 295 316 337 358 378 399 400
SACO	1 22 43 64 85 106 127 147 167 187 208 209 210 211 232 253 274 295 316 337 358 378 399 400

Table 4
Comparison of simulation results.

	Shortest path/cm	inflection points	iterations
ACO	32.73	6	–
IACO	29.80	10	–
SACO	28.63	5	12

Table 5
Optimal path result.

	Optimal Path Result
ACO	1 2 32 63 94 125 156 187 217 247 277 308 338 368 369 370 371 372 373 374 405 436 467 468 469 470 471 501 532 563 594 625 654 683 713 744 774 805 836 867 868 869 900
IACO	1 32 62 93 124 155 186 217 248 249 280 311 342 343 344 345 346 347 378 409 440 471 502 533 564 595 626 627 658 688 718 748 778 808 839 869 900
SACO	1 32 63 94 125 156 187 217 247 277 308 339 370 401 432 462 493 494 525 556 587 618 619 620 651 682 713 744 774 805 836 867 868 869 900

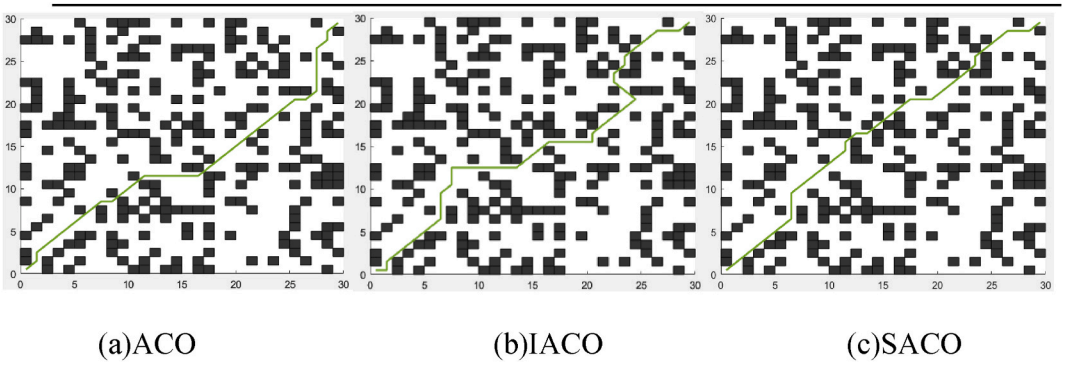


Fig. 8. Optimal planning path based on three ant colony algorithms.

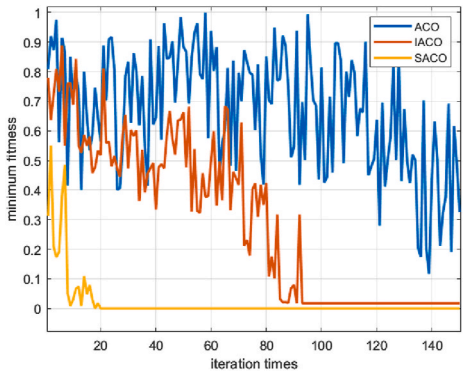


Fig. 9. Fitness iteration diagram.

Table 6
Comparison of simulation results.

	Shortest path/cm	inflection points	Runtime/s	iterations
ACO	50.28	17	13.5	–
IACO	45.11	12	9.8	93
SACO	43.94	12	6.3	20

dense, applying the ACO algorithm for robot path planning fails to effectively converge to the shortest path, resulting in a longer robot movement path. Applying the IACO algorithm for robot path planning requires more iterations to converge but tends to get stuck in local optima. On the other hand, applying the SACO algorithm results in a shorter robot movement path, and the algorithm can converge effectively. As shown in Table 8, the shortest path obtained with the SACO algorithm is 21.8% shorter than ACO and 9.8% shorter than IACO. The number of turning points is reduced by 58.3% and 41.1%, and the system's runtime is reduced by 51.1% and 40%, respectively. Therefore, the SACO algorithm can find the shortest movement path in a shorter time, and the path is smoother.

Table 7
Optimal path result.

	Optimal Path Result
ACO	1 41 81 121 161 201 242 283 324 365 406 447 487 528 529 530 531 492 493 494 495 536 577 618 659 700 741 782 823 864 905 946 986 1026 1067 1108 1148 1188 1228 1269 1310 1351 1352 1353 1394 1435 1476 1477 1438 1479 1520 1560 1600
IACO	1 41 81 121 161 201 242 283 324 365 406 447 487 528 529 530 531 492 493 494 495 536 577 618 659 700 741 782 823 864 905 946 986 1026 1067 1108 1148 1188 1228 1269 1310 1351 1352 1353 1394 1435 1476 1477 1438 1479 1520 1560 1600
SACO	1 42 83 124 125 126 167 208 249 290 291 332 373 414 455 496 536 577 618 659 700 741 782 823 864 905 946 987 1028 1069 1110 1151 1192 1233 1274 1275 1276 1316 1356 1397 1438 1479 1520 1560 1600

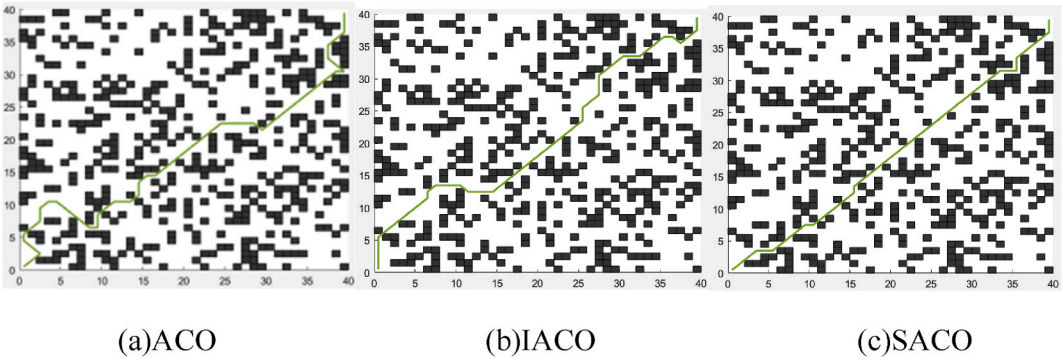


Fig. 10. Optimal planning path based on three ant colony algorithms.

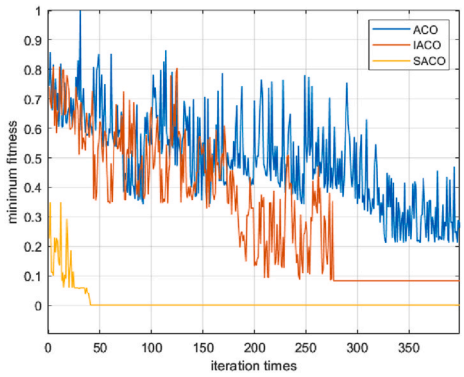


Fig. 11. Fitness iteration diagram.

Table 8
Comparison of simulation results.

	Shortest path/cm	inflection points	Runtime/s	iterations
ACO	74.32	24	22.3	–
IACO	64.42	17	18.3	277
SACO	58.08	10	10.9	40

5.4. Conduct comparative experiments with the A algorithm*

To validate the superiority of the SACO algorithm, comparative analysis experiments were conducted with three widely used A* algorithms on a 10 × 10 grid map. A 10 × 10 grid map was created, and robot path planning simulations were performed using A* with Manhattan Heuristic (A* M), A* with Euclidean Heuristic (A* E), A* with Chebyshev Heuristic (A* C), and the SACO algorithm,

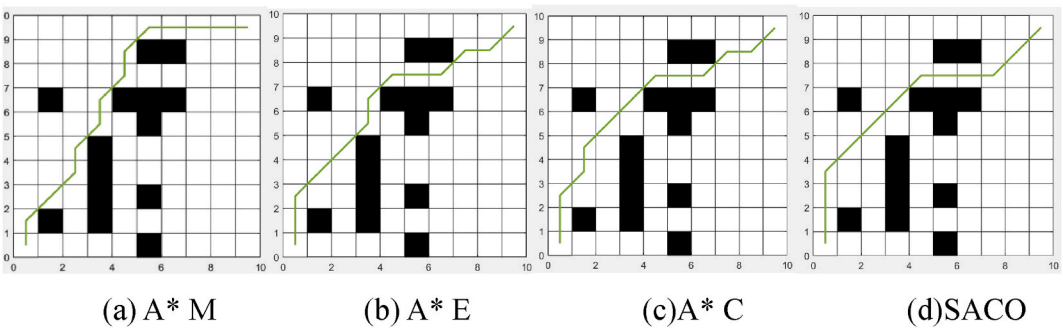


Fig. 12. Comparative experiment between SACO and three A* algorithms.

respectively.

From Fig. 12, it is evident that the A* algorithm, whether using Manhattan distance, Euclidean distance, or Chebyshev distance as a heuristic, struggles to achieve the optimal turning points. The improved algorithm in this paper has only 3 turning points, while A* with Manhattan Heuristic has 8 turning points, and A* with Euclidean Heuristic and A* with Chebyshev Heuristic both have 7 turning points. In terms of path length, SACO also outperforms A* with Manhattan Heuristic, and the improved algorithm exhibits better path smoothness compared to A* algorithms.

6. Conclusion

This paper addresses the issue of path safety in dam inspection by proposing SACO algorithm. This algorithm aims to provide secure and reliable path planning for AUV during dam inspection tasks, reducing energy consumption, enhancing task execution efficiency, while ensuring path smoothness and stability.

The SACO algorithm combines the strengths of traditional ACO, such as robustness, avoidance of local optima, and rapid computation speed. Additionally, it optimizes the smoothness of paths generated by the ACO. To reduce turning time and improve directional effectiveness in path selection, a corner-turning heuristic function is introduced. Experimental simulation results demonstrate that, within the same environment, SACO outperforms other algorithms in terms of path smoothness and the iterative stability of path planning.

Funding statement

This work is financially supported by Hubei Key Laboratory of Construction and Management in Hydropower Engineering (China Three Gorges University) Open Fund (No.2020KSD15), Natural Science Foundation of Hubei Province (No.2022CFC033) , and 2022 graduate course construction and cultivation project of China Three Gorges University (No.SDKC202203)

CRediT authorship contribution statement

Meng Ronghua: Writing – review & editing. **Cheng Xinhao:** Writing – original draft. **Wu Zhengjia:** Supervision. **Du xuan:** Funding acquisition.

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.

References

[1] Chua S.D.X., Xi L.X., Sediment load crisis in the Mekong river basin: severe reductions over the decades, Geomorphology (2022),419: 108484.
[2] L. Wang, C. Wu, L. Tang, et al., Efficient reliability analysis of earth dam slope stability using extreme gradient boosting method, Acta Geotechnica 15 (7) (2020), https://doi.org/10.1007/s11440-020-00962-4.
[3] XiaojieZhao Liu, QinLu ChaoyingZhang, ZhenhongYang ZhongLi, WuLiu-Zeng ChengshengZhu, LiquanLiu JingChen, Chuanjin.Integration of Sentinel-1 and ALOS/PALSAR-2 SAR datasets for mapping active landslides along the Jinsha River corridor, China, Eng. Geol. 284 (1) (2021).
[4] L.W.A.B. C, T.H. A, Y.Z. C, et al., The influence of fiber type and length on the cracking resistance, durability and pore structure of face slab concrete - ScienceDirect, Construct. Build. Mater. 282 (2023), https://doi.org/10.1016/j.conbuildmat.2021.122706, 07-26].
[5] F.F. Li, Y. Du, K.J. Jia, Path planning and smoothing of mobile robot based on improved artificial fish swarm algorithm, Sci. Rep. 12 (1) (2022) 659.
[6] J H Bai, Y J Oh, Global path planning of lunar rover under static and dynamic constraints[J], International Journal of Aeronautical and Space Sciences (2020) 1–9.
[7] S. Oh, M. Hahn, J. Kim, Simultaneous Localization and Mapping for Mobile Robots in Dynamic Environments[C]//2013 International Conference on Information Science and Applications (ICISA), IEEE, 2013, https://doi.org/10.1109/ICISA.2013.6579440.
[8] J. Meng, S. Wang, L. Jiang, et al., Accurate and efficient self-localization of AGV relying on trusted area information in dynamic industrial scene, IEEE Trans. Veh. Technol. (6) (2023) 72.
[9] S. Du, M. Ibrahim, M. Shehata, et al., Automatic license plate recognition (ALPR): a state-of-the-Art review, IEEE Trans. Circ. Syst. Video Technol. 23 (2) (2013) 311–325, https://doi.org/10.1109/TCSVT.2012.2203741.
[10] C. Miao, G. Chen, C. Yan, et al., Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm, Comput. Ind. Eng. (2021) 107230, https://doi.org/10.1016/j.cie.2021.107230.
[11] Z.Z.L. Song, An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve, Appl. Soft Comput. 100 (1) (2021).
[12] Zhu Z., Yin Y., Lyu H., Automatic collision avoidance algorithm based on route-plan-guided artificial potential field method, Ocean engineering (2023),271: 113737.
[13] Jiao J., Cheng J., Liu Y., et al., Inversion of TEM measurement data via a quantum particle swarm optimization algorithm with the elite opposition-based learning strategy, Comput. Geosci. (2023),174: 105334.
[14] a b Frantiek Duchoň, A.B.A. B, M.K.A. B, et al., Path planning with modified a star algorithm for a mobile robot - ScienceDirect, Procedia Eng. 96 (2014) 59–69.
[15] C Liu, L Wu, W Xiao, et al., An improved heuristic mechanism ant colony optimization algorithm for solving path planning[J], Knowledge-Based Systems 271 (2023) 110540.
[16] Input-to-state stability of stochastic Markovian jump genetic regulatory networks, Math. Comput. Simulat. (2023), https://doi.org/10.1016/j.matcom.2023.08.007.
[17] Harish Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl. Math. Comput. 274 (2016) 292–305, https://doi.org/10.1016/j.amc.2015.11.001.

[18] X. Liu, R. Deng, J. Wang, et al., COStar: a D-star Lite-based dynamic search algorithm for codon optimization, J. Theor. Biol. 344 (2014) 19–30, https://doi.org/10.1016/j.jtbi.2013.11.022.
[19] Analysis of markovian jump stochastic cohen-grossberg BAM neural networks with time delays for exponential input-to-state stability, Neural Process. Lett. (2023), https://doi.org/10.1007/s11063-023-11364-4.
[20] P. Zhenbing, C. Xuebo, Improved ant colony algorithm and its application in obstacle avoidance for robot, CAAI Transactions on Intelligent Systems (2015) 90–96.
[21] C. Miao, G. Chen, C. Yan, et al., Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm, Comput. Ind. Eng. (2021) 107230, https://doi.org/10.1016/j.cie.2021.107230.
[22] X.Y. Wang, L. Yang, Y. Zhang, et al., Robot path planning based on improved ant colony algorithm with potential field heuristic, Kongzhi yu Juece/Control and Decision 33 (10) (2018) 1775–1781, https://doi.org/10.13195/j.kzyjc.2017.0639.
[23] Y. Zhang, F. Wang, F. Fu, et al., Multi-AGV path planning for indoor factory by using prioritized planning and improved ant algorithm, J. Eng. Technol. Sci. 50 (4) (2018) 534–547, https://doi.org/10.5614/j.eng.technol.sci.2018.50.4.6.
[24] F Sui, X Tang, Z Dong, et al., ACO+ PSO+ A*: A bi-layer hybrid algorithm for multi-task path planning of an AUV[J], Computers & Industrial Engineering 175 (2023) 108905.
[25] L Wu, X Huang, J Cui, et al., Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot[J], Expert Systems with Applications 215 (2023) 119410.
[26] C. Ntakolia, D.V. Lyridis, A comparative study on Ant Colony Optimization algorithm approaches for solving multi-objective path planning problems in case of unmanned surface vehicles, Ocean engineering (Jul.1) (2022) 255.
[27] X. Xie, Z. Tang, J. Cai, The multi-objective inspection path-planning in radioactive environment based on an improved ant colony optimization algorithm, Progress in nuclear engergy (2022) 144.
[28] P He, G Jiang, S K Lam, et al., ML-MMAS, Self-learning ant colony optimization for multi-criteria journey planning[J], Information Sciences 609 (2022) 1052–1074.
[29] C. Cheng, Q. Sha, B. He, et al., Path planning and obstacle avoidance for AUV: a review, Ocean Engineering 235 (2023), https://doi.org/10.1016/j.oceaneng.2021.109355, 08-09].
[30] C Liu, L Wu, W Xiao, et al., An improved heuristic mechanism ant colony optimization algorithm for solving path planning[J], Knowledge-Based Systems 271 (2023) 110540.