# Application of Improved Ant Colony Algorithm Integrating Adaptive Parameter Configuration in Robot Mobile Path Design

Jinli Han
Department of Numerical Control Engineering, Shanxi Institute of Mechanical & Electrical Engineering, Changzhi, China

***Abstract*—Under the background of the continuous progress of Industry 4.0 reform, the market demand for mobile robots in major world economies is gradually increasing. In order to improve the mobile robot's movement path planning quality and obstacle avoidance ability, this research adjusted the node selection method, pheromone update mechanism, transition probability and volatility coefficient calculation method of the ant colony algorithm, and improved the search direction setting and cost estimation calculation method of the A* algorithm. Thus, a robot movement path planning model can be designed with respect to the improved ant colony algorithm and A* algorithm. The simulation experiment results on grid maps show that the planning model constructed in view of the improved algorithm, the traditional ant colony algorithm, the Tianniu whisker search algorithm, and the particle swarm algorithm designed in this study converged after 8, 37, 23, and 26 iterations, respectively. The minimum path lengths after convergence were 13.24m, 17.82m, 16.24m, and 17.05m, respectively. When the edge length of the grid map is 100m, the minimum planning length and total moving time of the planning model constructed in view of the improved algorithm, the traditional ant colony algorithm, the longicorn whisker search algorithm, and the particle swarm algorithm designed in this study are 49m, 104m, 75m, 93m and 49s, 142s, 93s, and 127s, respectively. This indicates that the model designed in this study can effectively shorten the mobile path and training time while completing mobile tasks. The results of this study have a certain reference value for optimizing the robot's movement mode and obstacle avoidance ability.**

***Keywords*—Ant colony algorithm; robots; mobile path planning; obstacle avoidance***

## I. Introduction

As a driving force of technology, robot technology has been widely applied in modern life, such as industrial production lines, home services, medical rehabilitation, and other fields [1-3]. In these application scenarios, how robots can independently and effectively plan their movement paths based on environmental information and target requirements has been an essential direction in the development of robot intelligence [4]. Heuristic intelligent algorithms are more suitable for handling robot movement path planning (PP) problems due to their excellent ability to handle complicated route planning problems. Ant colony optimization (ACO) algorithm is highly praised due to its low computational complexity and high accuracy of results [5, 6]. However, the parameter configuration in the computation process of ACO algorithm has an essential influence on the algorithm performance, and it is hard for manually determining suitable parameters to adapt to various environments. Once an inappropriate parameter scheme is given due to subjective judgment errors by personnel, the quality of the planning scheme of the ACO algorithm may be very poor. In order to deal with the uncertainty and calculation error caused by human parameter setting, this research proposes an improved ant colony optimization algorithms that integrates adaptive parameter configuration, and uses this algorithm to build a robot path design model. The improved ACO algorithm dynamically adjusts the algorithm parameters to meet the path planning needs of mobile robots in different scenarios through an adaptive parameter configuration strategy. Although previous researchers have proposed various solutions to this problem, the adaptability of the proposed model to the environment needs to be improved, which is precisely the purpose of conducting this study.

This study consists of four major sections. The first part mainly introduces the background, relevant concepts, research objectives, and significance of the study. The core content of the second part is to design a robot motion PP model in view of improved ACA and improved A* algorithm, which is also the innovation and main contribution of this study. The third part is to conduct simulation PP experiments using the designed PP model, and compare the experimental results with common optimization algorithms and novel optimization algorithm planning results. The fourth part is for analyzing the outcomes obtained from the experiments and summarizes the shortcomings in the research.

## II. Related Works

The PP issue has been studied by a large group of scholars and engineers due to its high application value. Xu et al. [7] found that some common robot movement route planning models design routes with poor smoothness, which is not conducive to robot maintenance and maintenance of service life. Therefore, the author team has designed a robot smooth PP method in view of improved particle swarm optimization (PSO) algorithm and Bessel transition curve. The simulation indicates that compared with traditional planning methods, the designed route is significantly smoother, and the total distance increase of the middle mobile is small. Li et al. [8] proposed a four-way search PP scheme suitable for mobile robots. This method achieves rapid optimization of PP by searching in four directions: horizontal and vertical. Compared with traditional

heuristic optimization algorithms, the four-way search scheme has strong advantages in solution space search, which helps to find high-quality paths with lower costs and satisfy various constraints. Yuan et al. [9] proposed a mobile PP algorithm for robots equipped with mobile sensor networks. Compared with traditional planning models, this algorithm has good adaptability and environmental awareness. The outcomes reveal that the algorithm in this study effectively improves the mobile PP ability of robots equipped with mobile sensor networks. The total length of the planned route is smaller than that of traditional planning models, and it has good collision avoidance ability. Liu and Jiang [10] proposed a PP model in view of the pigeon heuristic optimization algorithm. In the process of solving PP issues, this model takes the principle of avoiding collisions and unnecessary turns as much as possible to find a moving path. Through comparative experiments, it was found that this algorithm has superior PP performance in different scenarios. This indicates that its ability for finding the optimal path (OP) in complex environments exceeds other classical algorithms, providing an effective and efficient way to solve PP problems in complex scenarios. Meng et al. [11] proposed a safe and efficient LiDAR-based PP system to address the issue of insufficient obstacle avoidance ability in the PP of mobile robotic arms. This system is used to solve the navigation problem of four-wheel steering and four-wheel drive mobile robotic arms in manufacturing plants. In the study, the author utilized LiDAR technology to map the surrounding environment in real-time and identify obstacles. In addition, the study introduced a real-time collision avoidance algorithm to avoid dynamic and static obstacles. The simulation showcases that the PP system based on LiDAR proposed in the study exhibits high accuracy and robustness in handling navigation problems in manufacturing environments, and has excellent collision avoidance ability. Zhang et al. [12] presented a PP method for mobile robots in view of an improved local PSO algorithm. This algorithm increases the randomness and diversity in the local search process, improving the search capability. Through simulation experiments, the author verified the superior performance of this algorithm in mobile robot PP problems, indicating its good applicability and scalability in practical applications. This mobile robot PP method in view of improved local PSO algorithm provides an effective technical means to solve the navigation problem in complex environment in actual scenes.

In summary, although extensive research has been conducted to improve the planning performance and collision avoidance ability of intelligent PP models, there is little involvement in the construction of models that adjust adaptive parameters according to environmental characteristics. This mode is meaningful for improving the planning performance of PP models and the obstacle avoidance ability of robots.

## III. Robot Path Planning and Collision Avoidance Strategy in View of Improved ACO and A* Algorithms

The ACO algorithm was invented by imitating the foraging process of ants in nature, and has merits like strong adaptability and ease of utilizing in conjunction with other algorithms [13, 14]. For this reason, this algorithm was chosen in this study to construct a robot motion trajectory planning model. However, the ACO algorithm also has the disadvantage of being easily trapped in local optima, so it is also essential for enhancing the algorithm. In view of the shortcomings of the ACO algorithm, it can improve its node selection, pheromone update mechanism, transition probability calculation method and volatility coefficient calculation method.

### A. Design of Improved ACO Algorithm Based on Adaptive Parameter Setting

In actual working conditions, mobile robots will spend too much time in significant turning movements, and the corresponding energy loss will inevitably increase. Therefore, in PP, it is necessary to minimize significant turning points, improve the planning path, and diminish the motion cost of mobile robots. So now an improved planning path transition probability calculation method that integrates corner heuristic information is designed, as shown in Eq. (1).

$$P_{ij}^{(k)}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)\omega_{ij}^{\gamma}(t)}{\sum_{s\subset allowed_k}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)\omega_{ij}^{\gamma}(t)}, s\in allowed_k \\ 0, otherwise\end{cases} \quad (1)$$

The superscript of the variable in Eq. (1) represents the ant number; The subscript is the path number; $\tau_{ij}^{\alpha}(t)$, $\eta_{ij}^{\beta}(t)$ and $\omega_{ij}^{\gamma}(t)$ respectively represent the pheromone quantity, heuristic information and corner heuristic function on the corresponding track of the corresponding ant at time $t$; $allowed_k$ represents the feasible adjacency grid label set of ant $k$. The related calculating method of the corner heuristic function $\omega_{ij}(t)$ is depicted in Eq. (2).

$$\omega_{ij}(t)=\omega_1 T \quad (2)$$

In Eq. (2), $\omega_1$ is an adjustable parameter with a value range of (0,1); $T$ represents the turning cost. The corner of the path can be described as shown in Fig. 1, where the corners of paths (1) and (2) are both 45°, and the corners of paths (3) and (4) are 90° and 135°.

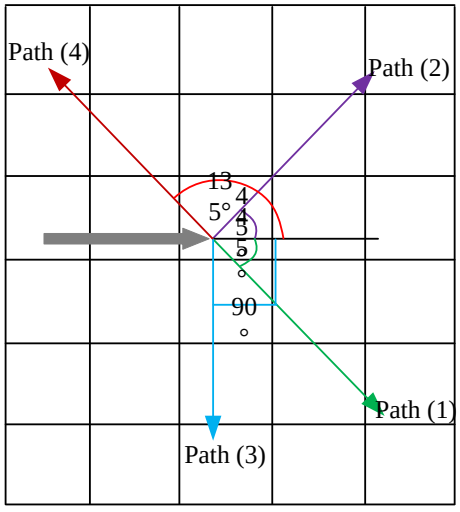



Fig. 1. Schematic diagram of improved path transition probability calculation method for path corners.

Therefore, $T$ in Eq. (2) can be calculated according to Eq. (3).

$$\begin{cases} T = 2, \theta \le 90° \\ T = 1, 90° \le \theta \le 180° \end{cases} \tag{3}$$

In the traditional ACO algorithm, the path is selected according to the roulette wheel method, and there is a relatively significant pheromone concentration in the algorithm operation phase, which leads to the slow convergence of the ACO algorithm in the calculation premise [15, 16]. Meanwhile, in very large environmental conditions, traditional ACO algorithms may experience convergence stagnation, which significantly reduces the global solution optimization performance of the algorithm [17]. Therefore, an adaptive parameter setting method is designed here, which uses a mixture of random sexual selection and deterministic selection to calculate the selected path, as shown in Formula (4).

$$s = \begin{cases} \arg\left(\max\left(\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}(t)\right]^{\beta} \cdot \left[\omega_{ij}(t)\right]^{\gamma}\right)\right), q \le q_0 \\ p_{ij}^{k}, else \end{cases} \tag{4}$$

In Eq. (4) $s$ represents the selected path; $p_{ij}^{k}$ is the possibility of the corresponding ant appearing on the corresponding path; $\alpha$, $\beta$ and $\gamma$ represent the corresponding ant numbers for $\tau_{ij}(t)$, $\eta_{ij}(t)$, and $\omega_{ij}(t)$, respectively; $q$ is a uniformly distributed random variable in the range of [0,1], and $q_0$ is the threshold corresponding to the deterministic selection. The calculation method for $q_0$ is shown in Eq. (5). In this new adaptive parameter setting method, random variable $q$ and deterministic selection threshold $q_0$ are used to show the randomness and certainty in the selection process respectively.

$$q_0 = \varepsilon \cdot \left\{0.2 + \left[\left(N_{\max} - N_c\right) / N_{\max}\right] \times 0.7\right\} \tag{5}$$

In Eq. (5), $\varepsilon$ is the adjustment coefficient; $N_{\max}$, $N_c$ serves as the maximum quantity of iterations and the current quantity of iterations. Combining the search characteristics of ant colony during the iteration process, the initial $q_0$ of the improved ACO algorithm is generally greater than $q$, which means that the initial path of the improved ACO algorithm is determined in a pseudo random probability manner. In the later stage, as the small value of $q_0$, ants are more likely to conduct random searches. It indicates that this strategy of selecting path nodes based on adaptive parameter calculation can effectively reduce the algorithm runtime, accelerate the algorithm convergence, and reduce the probability of stagnation. This increases the likelihood of the algorithm finding the optimal solution.

The traditional ACO algorithm will update the pheromone according to the way of updating all paths, but the disadvantage of this way is that the pheromone quantity of all paths may not differ greatly. This cannot highlight the competitiveness of the dominant path, and the convergence speed is slow. Therefore, to strengthen the attraction of the path, the pheromone updating method is now adjusted. The calculation method of pheromone $\tau_{ij}(t+n)$ at time $t+n$ is shown in Eq. (6).

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t, t+n) + \frac{h\left(L_a - L_n\right)}{L_n} \tag{6}$$

In Eq. (6), $L_a$ and $L_n$ are the optimal values in the iteration history and the optimal values in this iteration, respectively; $L_n$ is the adjustable coefficient. Therefore, when the calculation of each iteration of the algorithm is completed, when $L_a > L_n$, the corresponding path of this iteration is shorter. Eq. (6) will strengthen the pheromone strength of this iteration and save the OP generated in this iteration. On the contrary, if $L_a < L_n$, it means that the current path is not the shortest path. Eq. (6) will reduce the strength of pheromone.

Due to special terrain conditions, the constant volatility coefficient $\rho$ will diminish the likelihood of the algorithm finding the OP. When the value of $\rho$ is too large, although the algorithm converges faster, it also may fall into local optima; When the value of $\rho$ is too small, the likelihood of previously explored nodes being repeatedly explored will increase, and the convergence speed will also decrease. Therefore, it now adjusts the size of $\rho$ and obtains the value according to Eq. (7).

$$\rho = \begin{cases} 0.9\rho(t-1), L_A - L_n > b \\ 0.8\rho(t-1), a < L_A - L_n < b \\ 0.7\rho(t-1), L_A - L_n < a \end{cases} \tag{7}$$

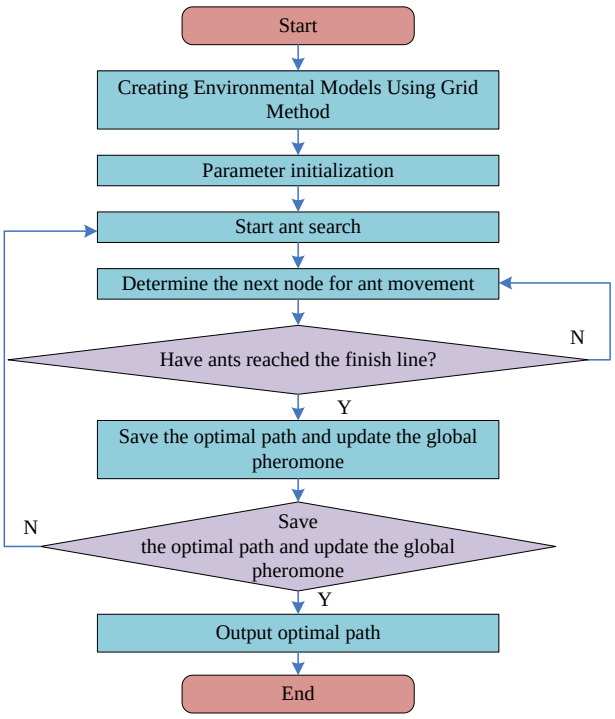



Fig. 2. Improved ACO algorithm calculation process.

In Eq. (7), $L_A$ is the worst case path in the current calculation result; $a$ and $b$ are two constants; The smaller the $L_A - L_n$ value, the greater the likelihood of it falling into local optima, and the faster the update speed of $\rho$. The improved ACO algorithm used in robot movement PP has been designed, and its calculation is shown in Fig. 2.

### B. Improved A* Algorithm and Robot Path Planning Model Design

Due to the fact that mobile robots often encounter moving obstacles in actual working environments, if the robot still follows the planned route, the time and energy consumption to reach the endpoint (EP) will increase, and even safety accidents may occur [18, 19]. Therefore, when designing a robot motion PP model, it is also necessary to consider obstacle avoidance issues. Robot movement PP cannot only consider the search efficiency of algorithms, but also the degree of twists and turns in the route. The classic A* algorithm is often applied to solve such problems, but it uses Manhattan distance and Euclidean distance to set heuristic functions, with only four search directions [20]. The fewer search directions in the classic A* algorithm will increase the number of probe nodes and the viewpoint of the path, which will affect the performance of the algorithm. Moreover, robots moving along winding paths can also waste too much time. Consequently, it is essential for enhancing the traditional A* algorithm by performing path search calculations in eight directions, and combining the Manhattan distance $h_M(n)$ and Euclidean distance $h_g(n)$ of the current point $n$ to design an estimated cost $h(n)$ that can simultaneously reduce the number of search nodes and bends. The calculation is demonstrated in Eq. (8).

$$h(n) = \max\left(abs\left(n_x - g_x\right), abs\left(n_y - g_y\right)\right) \quad (8)$$

In Eq. (8), $abs(\cdot)$ represents the absolute value operation; $g_x$ and $g_y$ represents the coordinates of the target node $g_y$ in both axis directions. Therefore, the improved A * algorithm calculation process is demonstrated in Fig. 3.

In this study, an environment model for robot motion is established, as shown in Fig. 4. The basic method for modeling is the grid method, which selects grids of appropriate size to simulate the environment. The static fault objects in the environment in Fig. 4 are simulated using a blue grid, with a green cross representing dynamic obstacles (DO), and static obstacles that suddenly enter the environment are described in red. The white grid in Fig. 4 shows the areas where the robot can move freely.

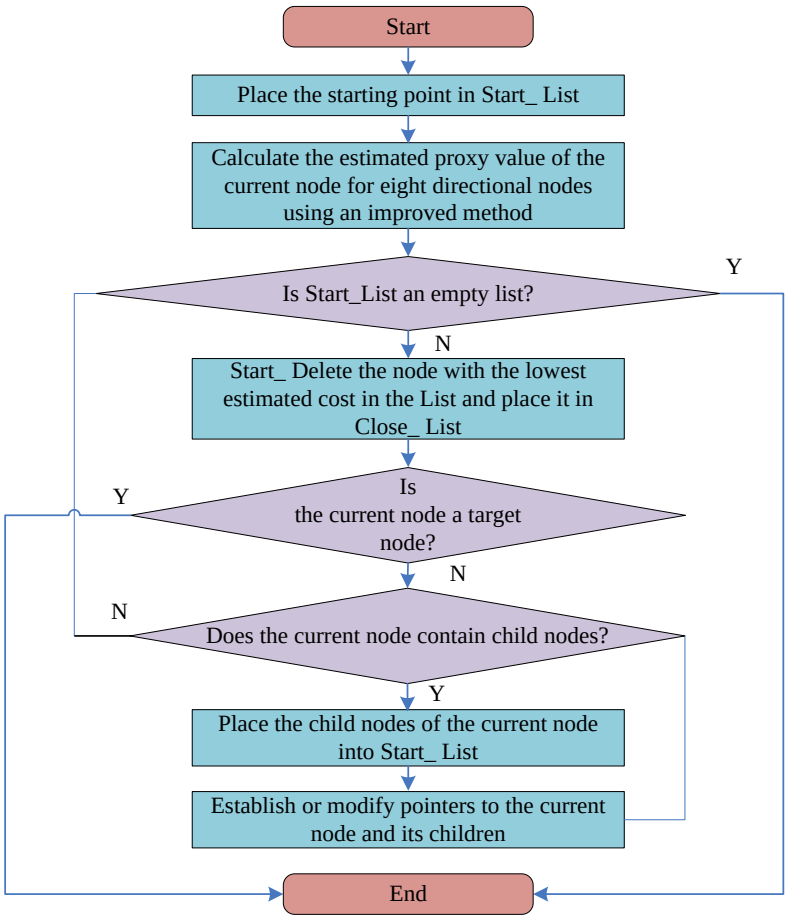



Fig. 3. Improved A* algorithm calculation process.

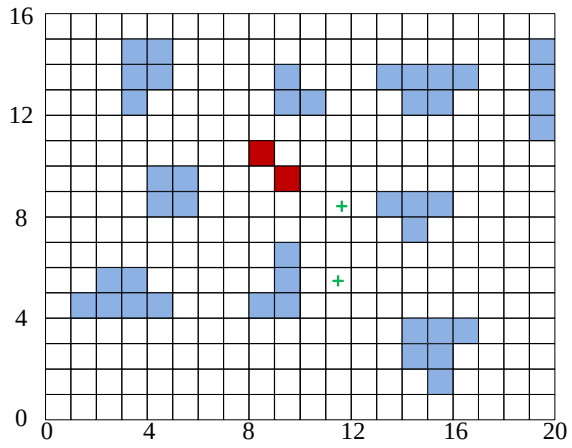



Fig. 4. Grid map of robot mobile environment simulation.

Robots need to continuously search for various obstacles within a limited range while moving, in order to make targeted obstacle avoidance behaviors. To perceive the movement path of the future DO, linear prediction models are now used to calculate the motion status of the DO. The current assumption is that the moving obstacle in the environment is $P(x, y)$, and the horizontal and vertical coordinate values in the orientation are correlated with time $t$. Therefore, $(x, y)$ can be calculated according to Eq. (9),

$$\begin{cases} x = at + b \\ y = ct + d \end{cases} \tag{9}$$

$g_y$ in Eq. (8) are unknown equation parameters. According to Eq. (9), calculate the corresponding $(x_l, y_l)$ (where $l = 1, 2, ..., n$) at $n$ time points to obtain a linear equation in matrix form, as shown in Eq. (10).

$$\begin{cases} X_n = T_n + A_m \\ Y_n = T_n + A_m \end{cases} \tag{10}$$

The matrix content in Eq. (10) is shown in Eq. (11).

$$\begin{cases} X_n = [x_1, x_2, ..., x_n]^T \\ T_n = \begin{bmatrix} t_1, t_2, ..., t_n \\ 1, 1, ..., 1 \end{bmatrix}^T \\ A_m = [a, b]^T \\ B_m = [c, d]^T \\ Y_n = [y_1, y_2, ..., y_n]^T \end{cases} \tag{11}$$

In this study, the error vector $E_n$ is set and calculated according to Eq. (12).

$$E_n = [e_1, e_2, ..., e_n]^T \tag{12}$$

Therefore, Eq. (13) holds.

$$E_n = X_n - T_n \cdot A_n \tag{13}$$

In Eq. (13), $A_n$ is the estimated value. The analytical error can be calculated according to Eq. (14).

$$J_n = \sum_{l=1}^{n} \lambda^{n-l} e_l^2 \tag{14}$$

The range of $\lambda$ values in Eq. (14) is (0,1). Its redefinition F is calculated according to Eq. (15).

$$P_{n-1} = \left(T_{n-1}^T T_{n-1}\right)^{-1} / \lambda \tag{15}$$

According to the observation values of moving obstacles at $n$ different times, their spatial coordinates can be calculated, thereby calculating the specific values of parameters $g_y$. The behavior of the robot sensors, which detect the position of the DO in the circumstance will accompany the entire movement of the robot, used to estimate parameters for corresponding updates. Finally, the subsequent position information of the obstacle can be obtained according to the above method.

In real-world application scenarios, not all obstacles present in the environment of mobile robots are stationary, and there may be obstacles that can move, such as humans. Therefore, robots also need to avoid DO, which requires higher control and PP capabilities. Below is the design of collision prediction and obstacle avoidance strategies for robots. Considering the sensor measurement capabilities and movement methods of most mobile robots on the market, it is assumed that robots use their own sensors to continuously measure the movement position, direction, and speed of surrounding obstacles. The motion directions of robots and DO are described in Fig. 5.

Based on Fig. 5, the collision avoidance strategy designed in this study is illustrated: if there is no method of collision between the robot and the DO, the motion trajectory of the two needs to be observed to determine if there is an intersection between the two. If there are no intersections, it is assumed that they will not collide. In this case, the robot does not need to take additional collision avoidance actions and can move according to the originally planned trajectory, as shown in Fig. 5 for A, B, and F. But in the case where there is an intersection point between the robot and the obstacle's motion trajectory, the two will collide, and at this point, it is necessary to redefine the robot's motion path. For example, in the C and D motion situations in Fig. 5, the obstacle intersects with the robot's motion at the side, and the corresponding collision avoidance strategy is for the robot to remain stationary. After the current motion trajectory of the DO does not intersect with the robot, the robot continues to move along the set path. In summary, to better address the problem of robot motion trajectory planning, it is necessary to combine the improved A* algorithm with the improved ACO algorithm. The corresponding computation is shown in Fig. 6.

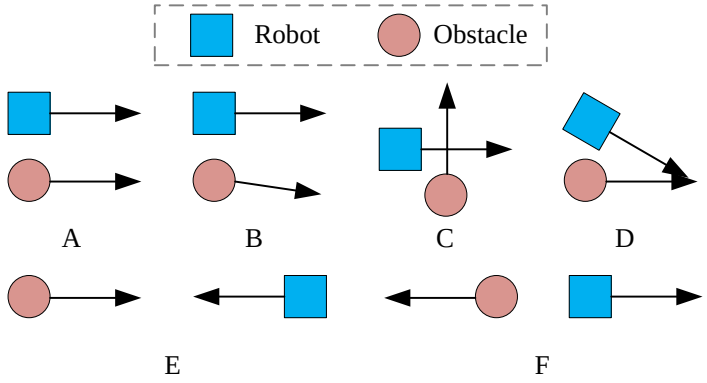



Fig. 5. Display diagram of dynamic obstacles and robot movement direction.

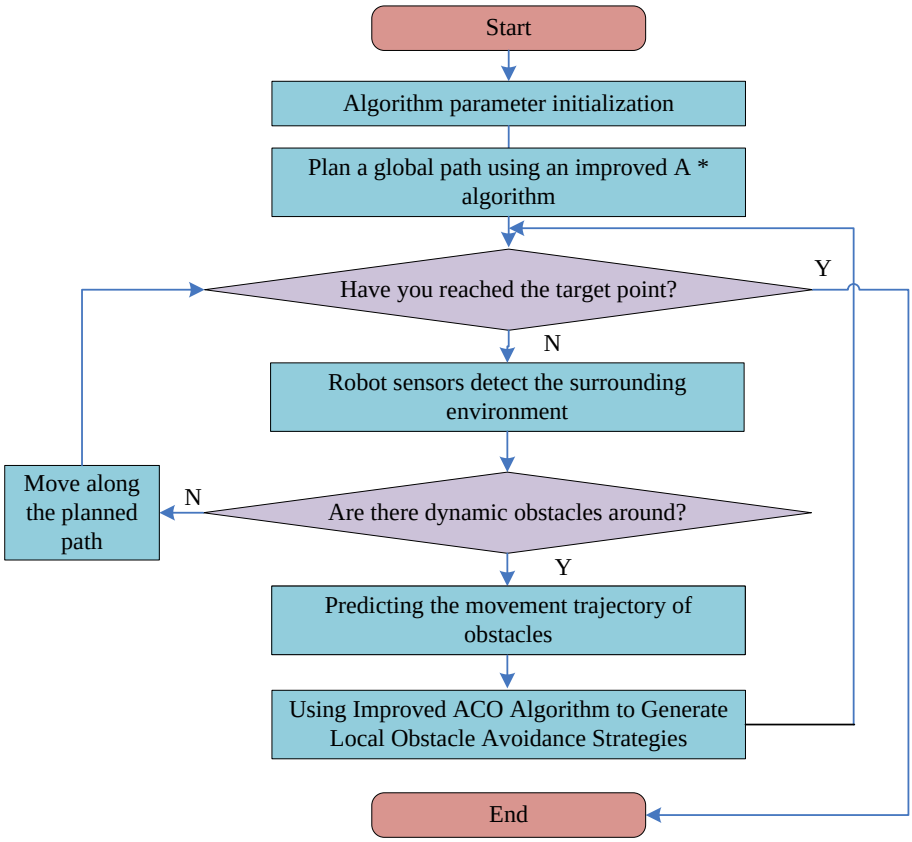



Fig. 6. Calculation process of robot path planning model using hybrid improved A * algorithm and IACO-A * algorithm.

## IV. Testing and Analysis of Robot Path Planning Model based on Hybrid Improved ACO Algorithm

After introducing a robot motion PP model in view of the improved ACO algorithm and the improved ACO-A * (IACO-A *) algorithm, the application of the model needs to be tested. In this study, only simulation experiments are used to verify the design model. Meanwhile, to simplify the experiment, the mobile environment in the test is abstraction into a grid map. The simulation experiment was run on the MATLAB2016 platform, and the parameters of the IACO-A * algorithm were determined through multiple trial runs as follows: the maximum number of cycles was 50, $\alpha = 4$, $\beta = 8$, $\rho = 0.7$, and the total quantity of ants was 50. The experimental circumstance is a two-dimensional map, and the minimum size of the tested grid is 10 m × 10 m, with a maximum size of 100m × 100m, with a grid growth step of 5m. In the experiment, the widely used ACA, PSO algorithm, and the novel Beetle Antennae Search (BAS) algorithm were selected to construct a comparative planning model. The parameters of the comparative model were also determined according to the trial operation method. The robot is set to move forward at a constant speed of 1m/s when no DO is encountered. After encountering obstacles and temporarily stationary, it first accelerates to 1m/s at an acceleration of 1m/s$^2$, and then continues to move at a constant speed. There are two DO in the

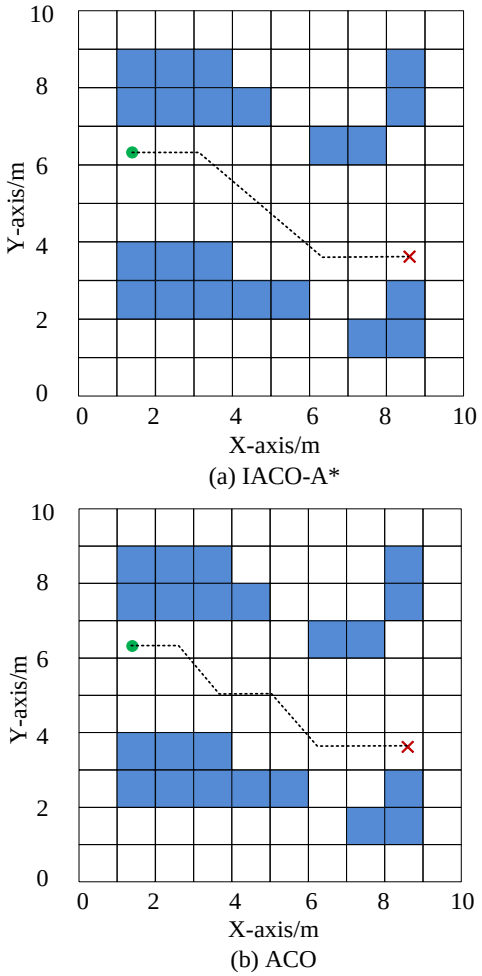
grid map, and their initial positions appear randomly. Referring to the actual working circumstance of mobile robots, the proportion of static obstacle grids to the total quantity of grids should be within the range of 20% to 70%; within this numerical range, the quantity of static obstacles generated is also random, and the setting results are shown in Table I.

TABLE I. COMPARISON OF ALGORITHM PARAMETER SETTING SCHEMES

| Number | Algorithm name | Parameter Name | Numerical value |
|---|---|---|---|
| #1 | ACO | Maximum number of cycles | 50 |
| | | Ant number | 200 |
| | | Walking distance | 1.5 |
| | | Pheromone volatilization factor | 0.6 |
| #2 | BAS | Maximum number of iterations | 50 |
| | | The initial length of the antennae of the longicorn beetle | 10 |
| | | Step decay factor | 0.95 |
| | | Maximum step size | 1.26 |
| | | Minimum step size | 0.35 |
| #3 | PSO | Maximum number of iterations | 50 |
| | | Inertia weight | 0.6 |
| | | Learning factor | 1.4 |
| | | Maximum speed | 27 |
| | | Minimum speed | 4 |

Firstly, a specific PP analysis is carried out using the planning results with a minimum grid size of 10 * 10 m and no DO conditions as a representative. The grid map generation results are randomly selected, and the OP planned by IACO-A * and ACO algorithms is indicated in Fig. 7. The horizontal and vertical axes in Fig. 7 represent the X and Y axes in the two-dimensional grid map, respectively; The scale units are all in meters; The blue grid represents the static obstacles that exist at the initial moment; The green dots and red crosses represent the starting point (SP) and EP of the robot's movement path, respectively; The black dashed line serves as the planned movement path. Fig. 7 demonstrates that the total path lengths of IACO-A * and ACO algorithms are 7.95 m and 12.64 m, respectively. Both the enhanced and the improved ACO algorithms can enable the robot to move from the SP to the EP; But the route planned by the IACO-A * algorithm has significantly fewer bends, and the overall route is smoother, resulting in a shorter motion time.

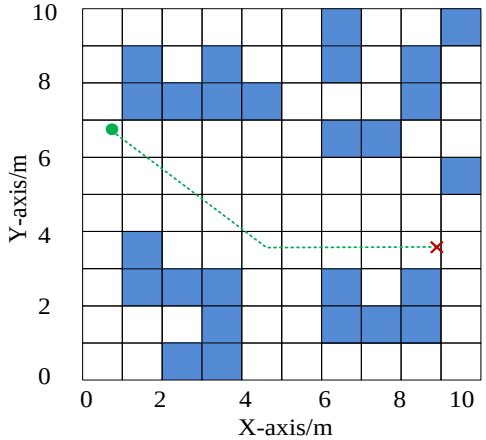
Further analysis of the planning results was conducted using 10 * 10 m DO conditions. Fig. 8(a), 8(b), 8(c), and 8(d) represent the initial planning path, the path when DO, O1 and O2 are added, the path to avoid O1, and the path to avoid O2, respectively. The green color in Fig. 8 serves as the path of the robot; Black serves as the movement path of DO; the dashed line serves as the planned path or the path that has been taken; The dotted line serves as the path of the obstacle or the robot during the avoidance process. Fig. 8(b) and 8(c) indicate that the robot will encounter side collisions with O1 DO during its movement. Therefore, choose to stay in place for a period of time until it is calculated that there is no longer an intersection between the two routes before continuing to move along the original route. Observing Fig. 8(d), it can be seen that after the robot detects a frontal collision with the DO O2, the IACO-A * algorithm generates local target points that modify the original path to some extent. After avoiding O2, the robot moves along the original path.



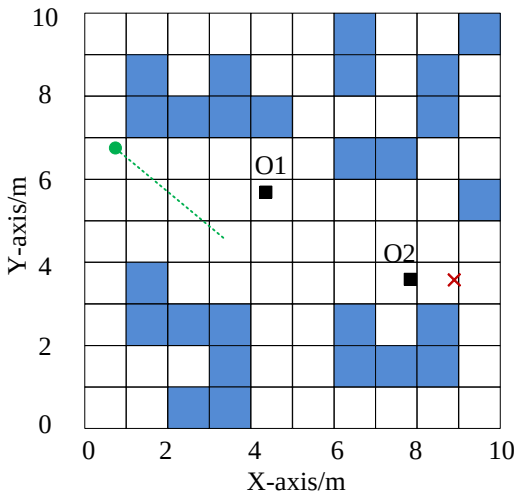
(a) IACO-A*

(b) ACO

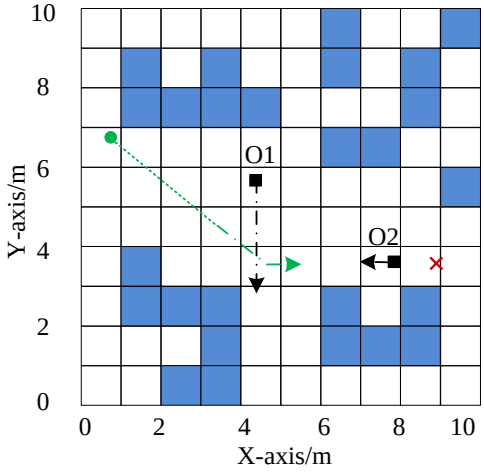
Fig. 7. IACO-A * and ACO algorithm in 10 × the optimal planning path in a 10 m grid map.



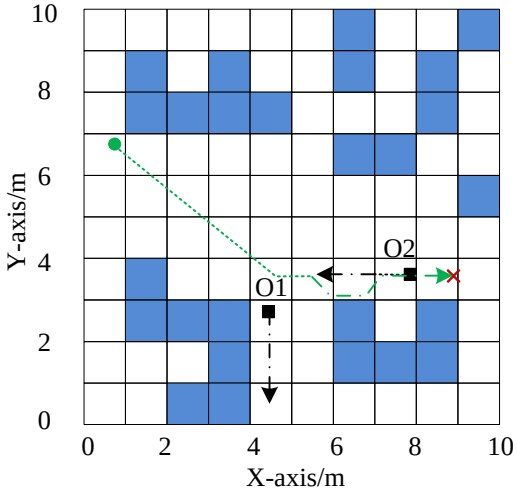
(a) Route planned by IACO-A* algorithm




(b) Environment when adding dynamic obstacles




(c) Path to avoid dynamic obstacle 1




(d) Path to avoid dynamic obstacle 2

Fig. 8. IACO-A * in 10 with dynamic obstacles × planned path in a 10m map.

It reanalyzes the planning outcomes of the unimproved ACO algorithm in the presence of DO, and the results are shown in Fig. 9. The meanings of the horizontal and vertical axes, icons, and lines in Fig. 9 are consistent with those in Fig. 8. Fig. 9 shows that although the route planned using the traditional ACO algorithm also avoids DO O1 and O2, the adjusted moving route significantly detours compared to Fig. 8. Based on Fig. 8 and 9, it is found that using the IACO-A * algorithm and the ACO algorithm, the overall travel distance of the two algorithms is 12.28 m and 16.74 m, respectively. This indicates that the IACO-A * algorithm exceeds the traditional ACO algorithm in the overall obstacle avoidance capability and route rationality of the robot PP.

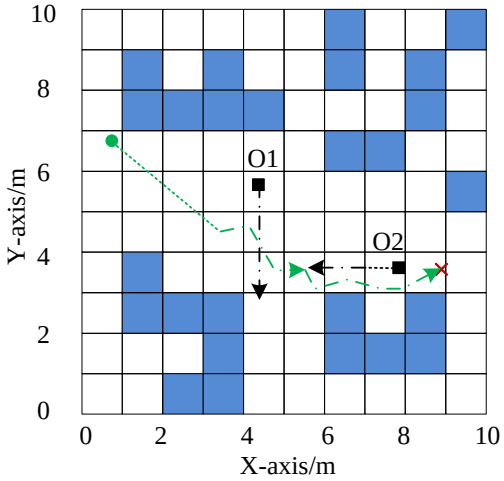



Fig. 9. ACO in 10 with dynamic obstacles × planned path in a 10 m map.

Next, it selects all the comparison models and calculates their minimum path length changes with the IACO-A * planning model during the training process of a 10 * 10m grid map with DO. It is indicted in Fig. 10. In Fig. 10, the horizontal axis (HA) serves as the quantity of iterations, while the vertical axis represents the minimum path length planned, in meters. Different subgraphs and line styles represent different planning models. Fig. 10 shows that the IACO-A *, ACO, BAS, and PSO planning models converge after 8, 37, 23, and 26 iterations, respectively. The minimum path lengths after convergence are 13.24m, 17.82m, 16.24m, and 17.05m, respectively. The experiment indicates that the PP model in this study based on the IACO-A * algorithm has the fastest convergence speed during training, and the total length of the OP after convergence is the smallest.

The performance of each planning model in the test dataset of a 10 * 10m grid map with DO is demonstrated in Fig. 11. The HA in Fig. 11 serves as repeated experimental tests, which are conducted to verify the stability of the output results of each planning model. Fig. 11 shows that when the number of repetitions is small, the minimum length fluctuation of the output routes of each planning model is more severe. However, as the number of repetitions increases, the minimum path length fluctuation gradually decreases. The minimum planning length standard deviations for IACO-A *, ACO, BAS, and PSO planning models are 0.82m, 1.24m, 0.95m, and 1.78m, respectively.

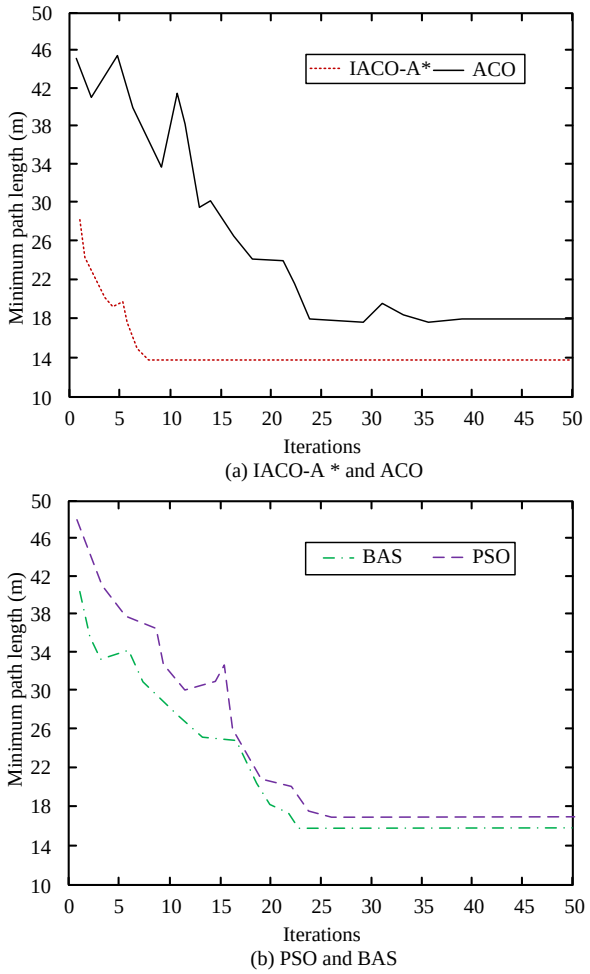



Fig. 10. Minimum path length of each planning model in training.

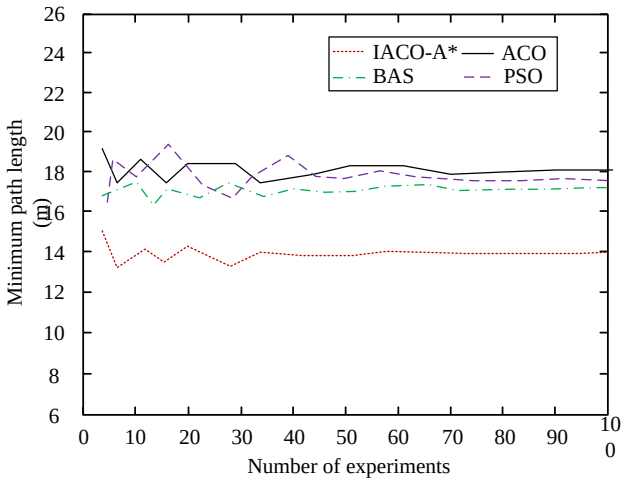



Fig. 11. Each planning model with dynamic obstacles 10 × Performance in 10 m grid map testing.

It further analyzes the minimum planning path length and corresponding movement time of each planning model under different grid size maps with DO. It is illustrated in Fig. 12. The HA in Fig. 12 serves as the edge lengths of different grid maps, in meters; The vertical axes of Fig. 12(a) and 12(b) represent the minimum travel path length and the corresponding total travel time, in units of m and s, respectively. The line style is used to distinguish the planning model. Fig. 12 serves as that as the size of the map grid grows, the time and total distance required for the robots to move from the SP to the EP in the map also increase. However, regardless of the size of the grid map, the minimum path length and total movement time of the IACO-A * planning model designed in this study are smaller than all the comparative planning models. For example, when the edge length of the grid map is 100m, the minimum planning length and total movement time of IACO-A *, ACO, BAS, and PSO planning models are 49m, 104m, 75m, 93m and 49s, 142s, 93s, and 127s, respectively.

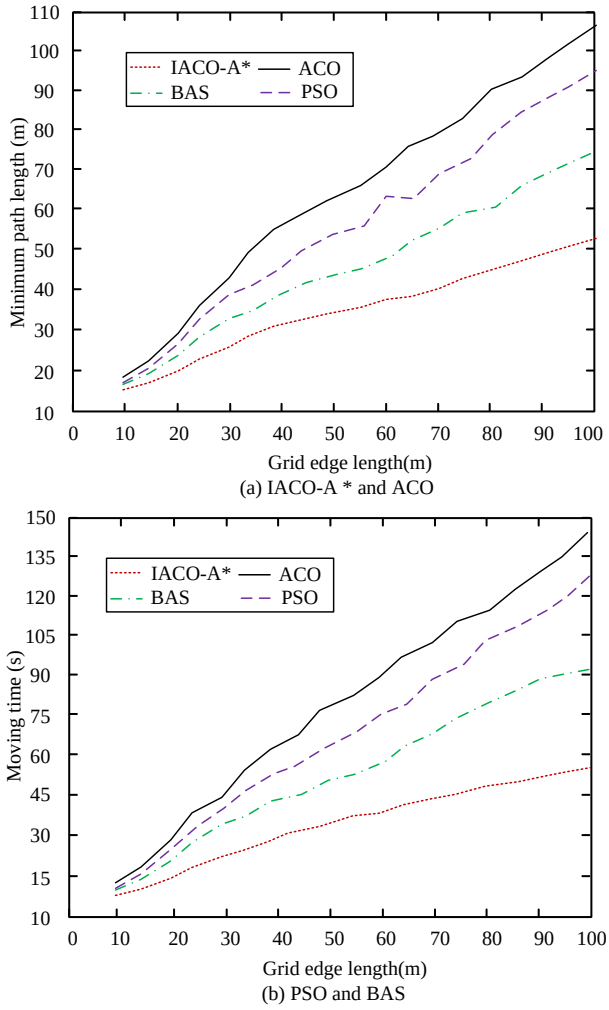



Fig. 12. Minimum path length and movement time of each model under different map sizes.

In order to further improve the reliability of the research results, the IACO-A * model is now placed in a domestic mobile robot product to carry out dynamic obstacle avoidance experiments in planar and curved scenes. The dynamic obstacles are spheres, rectangular parallelepipeds, and regular tetrahedrons, respectively. After repeated experiments, it was found that the robot system equipped with the IACO-A * model has the fastest dynamic obstacle avoidance response speed and the shortest obstacle avoidance time. Its practical

performance is significantly better than that of mobile robots equipped with other models.

## V. Results and Discussion

This study designed an improved ACO path planning model based on adaptive parameter setting, and conducted dynamic obstacle avoidance path planning experiments in a simulation grid map. The experimental results show that the model designed in this study has a grid size of 10 × the total planned path length under the conditions of 10 m and no dynamic obstacles is 7.95 m, which is lower than the output path length of the comparative model and can reach the set focus normally. Simultaneously, a in the case of a grid size of 10 * 10m and the presence of dynamic obstacles, the IACO-A * planning model corresponds to the robot using pause and generate local target points to avoid obstacles O1 and O2 that may cause side and frontal collisions, respectively. The ACO algorithm corresponds to the robot avoiding O1 and O2 obstacles with the same movement mode by generating local target points, but the overall movement distance is 16.74 meters, which is significantly higher than the data of the IACO-A * planning model. Moreover, in various scenarios where the grid edge length of the simulated grid map changes from 10 to 100, the total path length of the planning model output in this study is consistently lower than that of all the comparative models, and also lower than the total path length under the same conditions as in references [9] and [11]. Finally, the IACO-A * model was placed in a domestic mobile robot product to carry out dynamic obstacle avoidance experiments in planar and curved scenes. The dynamic obstacles have the shapes of sphere, rectangular cuboid, and regular tetrahedron, respectively. After repeated experiments, it was found that the robot system equipped with the IACO-A * model has the fastest dynamic obstacle avoidance response speed and the shortest obstacle avoidance time. Its practical performance is significantly better than that of mobile robots equipped with other models.

From the results of various types of experiments conducted in this study, it can be seen that the robot movement path planning model based on the improved ACO algorithm designed in this study has excellent path planning and obstacle avoidance capabilities, which can enable the robot to smoothly and efficiently avoid dynamic obstacles in the route.

## VI. Conclusion

In this study, an improved ACO algorithm and an improved A * algorithm were developed, and the two were fused to construct a robot motion PP model. The experiment reveals that under the conditions of grid size 10 * 10m and no DO, the total path lengths of IACO-A * and ACO algorithms are 7.95m and 12.64m, respectively. Both the enhanced and the improved ACO algorithms can enable the robot to move from the SP to the EP. Under the conditions of a grid size of 10 * 10 m and the presence of DO, the IACO-A * planning model corresponds to robots that use pause and generate local target points to avoid obstacles O1 and O2 that may cause side and frontal collisions, respectively. The ACO algorithm corresponds to the robot that avoids obstacles O1 and O2 with the same motion mode by generating local targets; however, the overall mobile distance of 16.74m is significantly higher than the data of the IACO-A * planning model. During the training process, the IACO-A *, ACO, BAS, and PSO programming models completed convergence after 8, 37, 23, and 26 iterations, respectively. The minimum path lengths after convergence were 13.24m, 17.82m, 16.24m, and 17.05m, respectively. When the edge length of the grid map is 100m, the minimum planning length and total movement time of the IACO-A *, ACO, BAS, and PSO planning models are 49m, 104m, 75m, 93m and 49s, 142s, 93s, and 127s, respectively. The experimental data prove that the robot motion PP model designed in this study with respect to the improved ACO algorithm has excellent PP and obstacle avoidance capabilities. However, due to research limitations, it was not possible to invite multiple industry experts to subjectively evaluate the practicality of the model. From the results of this experiment, it can be seen that the shape of obstacles has an extremely small impact on the path planning results of the IACO-A * model.

## Acknowledgment

The research was supported by 2020 Scientific and Technologial Innovation Programs of Higher Education Institution in Shanxi: “Research and Development of Industrial Robot Trajectory Planning Algorithm Optimization and Experimental Device (No.: 2020L0760)”.

## References

[1] Y. Li, W. Wei, Y. Gao, D. Wang, and Z. Fan, “PQ-RRT*: An improved path planning algorithm for mobile robots,” Expert Syst. Appl, vol. 152, pp. 113425, April 2020.

[2] Y. Liu, Y. Li, Y. Niu, and D. Jin, “Joint optimization of path planning and resource allocation in mobile edge computing,” IEEE Trans. Mob. Comput, vol. 19, no. 9, pp. 2129-2144, June 2020.

[3] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. S. Shu, “Path planning for UAV-mounted mobile edge computing with deep reinforcement learning,” IEEE Tran. Veh. Technol, vol. 69, no. 5, pp. 5723-5728, March 2020.

[4] X. Zhang, J. Lai, D. Xu, H. Li, and M. Fu, “2D Lidar-based SLAM and path planning for indoor rescue using mobile robots,” J. Adv. Trans, vol. 2020, no. Pt.9, pp. 8867937, November2020.

[5] Y. H. Hsu and R. H. Gau. “Reinforcement learning-based collision avoidance and optimal trajectory planning in UAV communication networks,” IEEE Trans. Mob. Comput, vol. 21, no. 1, pp. 306-320, June 2020.

[6] F. O. Coelho, M. F. Pinto, and J. Souza, “Hybrid methodology for path planning and computational vision applied to autonomous mission: A new approach,” Robotica, vol. 38, no. 6, pp. 1000-1018, July 2020.

[7] L. Xu, M. Cao, and B. Song, “A new approach to smooth path planning of mobile robot based on quartic Bezier transition curve and improved PSO algorithm,” Neurocomputing, vol. 473, no. Feb.7, pp. 98-106, February 2022.

[8] K. Li, C. Yuan, J. Wang, and X. Dong, “Four-direction search scheme of path planning for mobile agents,” Robotica, vol. 38, no. 3, pp. 531-540, 2020.

[9] W. Yuan, N. Ganganath, C. T. Cheng, G. Qing, C. M. Lau, and Y. Zhao, “Path-planning-enabled semiflocking control for multitarget monitoring in mobile sensor networks,” IEEE Trans. Ind Inform, vol. 16, no. 7, pp. 4778-4787, July 2020.

[10] A. Liu and J. Jiang, “Solving path planning problem based on logistic beetle algorithm search–pigeon-inspired optimisation algorithm,” Electron Lett, vol. 56, no. 21, pp. 1105-1108, 2020.

[11] J. Meng, S. Wang, Y. Xie, G. Li, X. L. Zhang, L. Jiang, and C. Liu, “Safe and efficient navigation system for 4WS4WD mobile manipulator in manufacturing plants,” Meas. Sci. Technol, vol. 32, no. 4, pp. 45203, November 2020.

[12] L. Zhang, Y. Zhang, and Y. Li, "Mobile robot path planning based on improved localized particle swarm optimization," IEEE Sens J, vol. 21, no. 5, pp. 6962-6972, November 2020.

[13] M. Wahab, S. Nefti-Meziani, and A. Atyabi, "A comparative review on mobile robot path planning: Classical or meta-heuristic methods?" Annu. Rev. Control, no. 50, pp. 233-252, October 2020.

[14] X. Deng, R. Li, L. Zhao, K. Wang, and X. Gui, "Multi-obstacle path planning and optimization for mobile robot," Expert Syst. Appl, vol. 183, pp. 115445, November 2021.

[15] J. Zan, "Research on robot path perception and optimization technology based on whale optimization algorithm," J. Comput. Cognitive Eng, vol. 1, no. 4, pp. 201-208, March 2022.

[16] S. Wu, Y. Du, and Y. Zhang, "Mobile robot path planning based on a generalized wavefront algorithm," Math. Probl. Eng, vol. 2020, no. Pt.6, pp. 6798798, March 2020.

[17] L. Tiseni, D. Chiaradia, M. Gabardi, M. Solazzi, D. Leonardis, and A. Frisoli, "UV-C mobile robots with optimized path planning: Algorithm design and on-field measurements to improve surface disinfection against SARS-CoV-2," IEEE Robot. Autom. Mag, vol. 28, no. 1, pp. 59-70, January 2021.

[18] T. Hossain, H. Habibullah, R. Islam, and R. V. Padilla, "Local path planning for autonomous mobile robots by integrating modified dynamic-window approach and improved follow the gap method," J. Field Robot, vol. 39, no. 4, pp. 371-386, December 2021.

[19] J. Li, J. Sun, L. Liu and J. Xu, "Model predictive control for the tracking of autonomous mobile robot combined with a local path planning:," Meas. Control, vol. 54, no. 9/10, pp. 1319-1325, October 2021.

[20] W. Chi, Z. Ding, J. Wang, G. Chen, and L. Sun, "A generalized voronoi diagram based efficient heuristic path planning method for RRTs in mobile robots," IEEE Trans. Industrial Electron, vol. 69, no. 5, pp. 4926-4937, May 2021.