

A novel A* method fusing bio-inspired algorithm for mobile robot path planning

Yang Sun^{1,*} and Haipeng Wang²

¹Software College, Shenyang Normal University, Shenyang 110034, China

²SIASUN Robot & Automation CO.,Ltd.

Abstract

The path planning of mobile robot is to find an optimal collision-free path in time distance or space from the starting point to the target point in a given environment. With the popularization and application of mobile robots, if the efficiency of mobile robots path is not high, the working quality will be seriously affected. How to quickly plan an effective safe path is of great research significance and practical application value. Therefore, we propose a novel A* algorithm based on Bio-inspired algorithm for mobile robot path planning. Firstly, the synchronous bidirectional A* algorithm is used to optimize the pheromone of ant colony algorithm, and the transition probability and pheromone update mechanism of ant colony algorithm are improved, so that the global optimization speed of the algorithm is faster and the path length of mobile robot is shortened. Furthermore, the static path is used to initialize the pigeon algorithm. Then, the improved pigeon algorithm is utilized to plan the local path of the mobile robot, and the simulated annealing criterion is introduced to solve the local optimal problem. The logarithmic S-type transfer function is adopted to optimize the step size of the pigeon number, so that the collision with the dynamic obstacles can be better avoided. Finally, a modified B-spline curve is used to smooth and re-plan the path. The simulation results show that the proposed method can realize path planning more effectively in complex dynamic environment.

Keywords: mobile robot path planning, A* algorithm, pigeon algorithm, ant colony algorithm, B-spline curve.

Received on 27 August 2021, accepted on 07 September 2021, published on 14 September 2021

Copyright © 2021 Yang Sun *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [Creative Commons Attribution license](#), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.14-9-2021.170953

*Corresponding author. Email: snowycry@qq.com

1. Introduction

Due to the characteristics of autonomous operation and flexible movement, mobile robots have a broad prospect of development and utilization in important fields such as national defense science and technology, life service, production and construction [1,2]. The path planning of mobile robot mainly realizes the followings: (1) make the

robot arrive at the predetermined target point from the starting point; (2) Obtain the collision-free safe path; (3) Obtain the optimal or sub-optimal distance path and optimize the path at the same time. In general, path planning is divided into global path planning [3,4] and local path planning based on completely known, completely unknown or partially known environmental information. In the static known environment, there are some methods such as Theta* [5], Lazy Theta* [6], etc.,

and in the dynamic unknown environment, there are some methods such as Field D* [7], etc.. Field D* algorithm uses path interpolation to smooth the path, but it has the problems of large amount of computation and resource consumption. On the other hand, Theta* introduces sight line algorithm on the basis of A* algorithm, which greatly optimizes the path obtained by A* algorithm, and further proposes Lazy Theta* algorithm, which reduces the calculation consumption time on the basis of Theta*, thus obtaining the path with faster calculation and shorter path length.

Existing path planning mainly focuses on global path planning [8]. However, when sudden threats are found in the movement of mobile robots, it is also urgent to solve the problems to plan local paths, reduce the path length and evaluation function value to improve the path performance. Researchers have done a lot of work in this field. Common path planning algorithms include ant colony algorithm, A* algorithm (2) and artificial potential field method, etc.. Corno [9] proposed a hop-point search method to improve algorithm A*, reduced the search of unnecessary nodes and improved the search efficiency. Wang [10] proposed to introduce the weight of exponential decay into the heuristic function of algorithm A*, so as to reduce the number of search nodes and improve the search efficiency of the algorithm. Hu [11] introduced the adaptive weight coefficient into the map operator part of the pigeon swarm algorithm to improve the search efficiency of the pigeon swarm algorithm. Huang [12] reduced the search range of A* algorithm through constraint function and improved the efficiency of the algorithm. Du [13] used the probability selection mechanism to distinguish map operator and landmark operator, which improved the global search ability of the algorithm. Although the above algorithm solves the problems of slow convergence speed and easily falling into local optimal solution to a certain extent, there are still problems such as uneven path and failure to consider unexpected obstacles.

Grid method is a classic method in environment modeling. The working environment of mobile robots is cut into different grids with the same size and connected with each other, and each grid corresponds to the corresponding position information. In reference [14], the improved A* algorithm was successfully used to realize path planning in static environment. Dijkstra algorithm is a breadth-first search algorithm. Tan [15] adopted the improved Dijkstra algorithm to realize global path planning, but its local path search was slow. It was difficult to quickly plan a smooth and safe optimal path. Wang [16] adopted improved ant colony algorithm to realize global path search, but the data storage capacity in the search process was large. Qu [17] successfully expanded the search neighborhood and realized collision-free trajectory planning quickly. References [18,19]

introduced an improved multi-order B-spline curve for path smoothing processing, but this method could not complete the smoothing processing of dynamic path, and the running process was cumbersome and inefficient. When the environment changes or it is affected by other external factors, the established path will be invalid. Therefore, it is urgent to enhance the timeliness of path and solve the problem of path planning in dynamic environment. Literature [20] proposed a multi-level hybrid algorithm combining A* algorithm and artificial potential field method to realize the path planning in dynamic environment. However, the artificial potential field method could not well plan the local shortest path in the application of local path planning, which reduced the overall operating efficiency [21,22].

The advantages of ant colony algorithm with positive feedback and parallelism make it more suitable for robot path planning and other problems [23]. The pigeons swarm algorithm is favored by researchers because of its excellent ability in UAV route planning and other problems. This paper presents a new A* algorithm for robot path planning based on the combination of improved ant colony(ACO) and improved pigeon-inspired optimization (PIO). Firstly, the improved A*-ACO algorithm is used for global static path planning. Secondly, the global static path is applied to the initialization process of the PIO, so that the algorithm can obtain more initial information. On the basis of the global path, the improved PIO is used for local dynamic obstacle avoidance, so that the mobile robot can bypass the dynamic obstacle for local path planning. Finally, a modified B-spline curve is used to smooth and re-plan the path.

This paper is organized as follows. In section 2, the global planning algorithm with improved ACO is introduced. Section 3 illustrates the local planning algorithm with improved PIO. We give the analysis for the experiment in section 4. The conclusion is given in section 5.

2. Global planning algorithm

2.1. Improved A* algorithm

A* algorithm was used by reference [24]. It is a heuristic path planning algorithm. In this paper, the improved A* algorithm is used to optimize the initial pheromone of the ant colony algorithm and reduce the search time, as shown in figure 1.

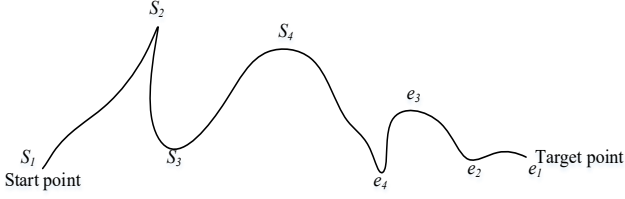


Figure 1. Flow chart of searching

In practice, the previous node of the current node in both directions is used as the target node for the opposite searching direction. For example, the current node S_3 in the forward search is searched by the node e_2 in the reverse search as the target node. In the reverse search, the current node e_4 is searched by the node S_3 in the forward search as the target node. This ensures that the target nodes searched in both directions are updated simultaneously. In order to effectively improve the efficiency of the algorithm, the obtained optimized path is denoted as R , and its initial pheromone is set as:

$$\tau(R) = kC; \quad k > 1 \quad (1)$$

Where k is the coefficient greater than 1. C represents the initial pheromone on the other path.

2.2. Direction evaluation-based heuristic information

In this paper, direction evaluation is introduced into heuristic information based on the improved A* algorithm. Before using the evaluation function $f(n)$ to calculate the cost of each extension node, the direction evaluation function $D(n)$ of the extension node is constructed. The direction evaluation function $D(n)$ is used to screen the extended nodes, and the nodes searching toward the target point are retained, and the nodes searching away from the target point are discarded to improve the search speed. In here, the direction evaluation function is:

$$D(n) = \cos \theta \quad (2)$$

Where $D(n)$ is the direction evaluation function of the extended node N . θ is the angle between the vector formed by the extension node and its parent node and the vector formed by the target point and the starting point. In this case, the coordinates of the starting point and the target point are known. Let (x_1, y_1) be the coordinate of the vector composed of the target point and the starting point. (x_2, y_2) is the coordinate of the vector formed by the extension node and its parent node. Then the angle between two vectors can be expressed as:

$$\cos \theta = \frac{x_1 \cdot x_2 + y_1 \cdot y_2}{\sqrt{x_1^2 + y_1^2} \cdot \sqrt{x_2^2 + y_2^2}} \quad (3)$$

The coordinate of the parent node $N-1$ is (x_{n-1}, y_{n-1}) .

The coordinate of the extension node N is (x_n, y_n) . The coordinate of the starting point and the target point are (x_{start}, y_{start}) and (x_{end}, y_{end}) . The coordinates of the two vectors are:

$$\begin{cases} x_1 = x_{end} - x_{start} \\ y_1 = y_{end} - y_{start} \\ x_2 = x_n - x_{n-1} \\ y_2 = y_n - y_{n-1} \end{cases} \quad (4)$$

In this paper, an eight-neighborhood node extension is adopted, when $D(n) \geq 0$, these nodes will be retained and added to the OPEN table for subsequent calculation. When $D(n) < 0$, these nodes are ignored to speed up the search.

2.3. Improved ant colony algorithm

The traditional ant colony algorithm conducts search operation through pheromone guidance [25]. In this paper, random selection mechanism is introduced to increase the diversity of solutions based on the transition probability of traditional ant colony algorithm. Let q_0 be the random selection factor. $q_0 \in (0,1)$. $q_1 \in [0,1]$ is a constant. When $q_0 \leq q_1$, after eliminating the obstacle nodes, it randomly selects any node around the current node as a feasible node; Otherwise, the feasible nodes are selected according to the probability transfer formula. The details are as follows:

$$S = \begin{cases} rand(allowed_k) & q_0 \leq q_1 \\ \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in allowed_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta} & q_0 > q_1 \end{cases} \quad (5)$$

Where α is the pheromone heuristic factor. β is the heuristic information factor. $\tau_{ij}(t)$ is the pheromone concentration connecting vertex i and j at the beginning of the t -th iteration. $\eta_{ij}(t)$ is heuristic information. $rand(allowed_k)$ means to randomly select the next node in a set of grids that are allowed to be selected.

To avoid the ant colony algorithm falling into local optimum affected by the non-optimal path pheromone interference. In this paper, we introduce reward and penalty factors into pheromone updating strategy. That is, the optimal path L_b after the current iteration is enhanced with its pheromone concentration. The pheromone concentration of the worst path L_w after the current iteration is reduced. Moreover, the local optimum should be jumped out at the beginning of iteration. In the later iteration, the influence of reward and punishment factors should be gradually reduced, so the trigonometric function is introduced as the coefficient. The updated formula after the above improved operation is as follows:

$$\tau_j(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^{(k)}(t) + b \quad (6)$$

$$b = \begin{cases} a[\cos((\pi/N_c)+1)/2]L_b, & \text{the current is optimal} \\ -a[\cos((\pi/N_c)+1)/2]L_w, & \text{the current is worst} \end{cases} \quad (7)$$

$$\tau_{ij}^{(k)}(t) = Q/L_k \quad (8)$$

Where b is the reward and punishment factor. ρ is pheromone volatile factor. $\Delta\tau_{ij}^{(k)}(t)$ is the pheromone increment on edges i and j . m is colony size. a is the adjustable coefficient, and the weight of reward and punishment factors can be adjusted according to the actual situation. N_c is the maximum number of iterations. Q is the pheromone intensity coefficient. L_k is the path length of the k -th ant in this iteration.

3. Local planning algorithm

3.1. Path evaluation function

The evaluation function of the robot path is as follows:

$$f = kf_1 + (1-k)f_2 \quad (9)$$

Where f represents the evaluation function. f_1 represents threat evaluation function. f_2 represents the path evaluation function. $0 \leq k \leq 1$ is the weight ratio between f_1 and f_2 . f_1 is the threat evaluation function of the robot, and the road sections are divided into 10 parts. Five symbolic points on the road section are taken to represent the threat cost of the path in this section, then the threat cost of the road section i is:

$$f_1 = L_i^5 / 5 \sum_{k=1}^N t_k \left(\frac{1}{d_{0.1,k}^4} + \frac{1}{d_{0.3,k}^4} + \frac{1}{d_{0.5,k}^4} + \frac{1}{d_{0.7,k}^4} + \frac{1}{d_{0.9,k}^4} \right) \quad (10)$$

Where L_i is the length of the road section. t_k is the threat factor, representing the influence degree of obstacles. N is the number of obstacles. $d_{0.5,k}$ is the distance between the point $5L_i/10$ and the k -th obstacle.

f_2 is the path evaluation function received by the mobile robot, which is improved by referring to the evaluation function in algorithm A* as follows:

$$f_2 = \sum_{i=1}^n (d_{i,i+1} + d_{i+1,e}) \quad (11)$$

Where $d_{i,i+1}$ represents the distance between node i and node $i+1$. $d_{i+1,e}$ represents the distance between node $i+1$ and target point e .

3.2. Initialization of Pigeon Algorithm

Based on the special navigation behavior of pigeons in the homing process, Duan et al. [26] proposed a bionic swarm intelligent optimization algorithm (pigeon swarm optimization algorithm). Pigeon algorithm is mainly used in UAV route planning and other fields. However, the existing Pigeon algorithm does not initialize the initial path during route planning, which leads to the low execution efficiency of the final algorithm. Based on the excellent ability of pigeon algorithm, this paper applies the improved pigeon algorithm to the path planning of robot. The path obtained in the static planning stage can be regarded as a relatively excellent solution set, which can be used as the initial population of the pigeon algorithm to improve the execution efficiency of the algorithm.

When a dynamic obstacle is detected, the static path between the local starting point and the local target point is set in the static stage to initialize the pigeon swarm algorithm. The details are shown in figure 2.

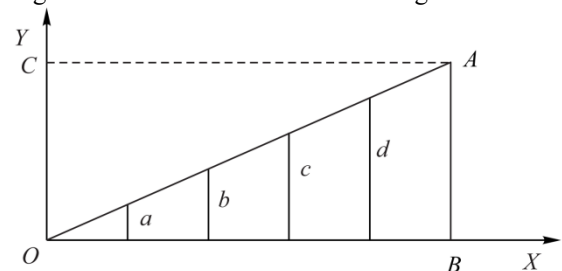


Figure 2. Schematic diagram of initialization

In the actual calculation, this paper simplifies the calculation by fixing the abscissa. Point O and point A in figure 2 are local starting points and local target points

respectively, and OA is divided into n equal parts along the horizontal direction. $a \sim d$ represents $n-1$ bisectors.

Through the above operation, the initial population of the pigeon algorithm is composed of initial data with normal distribution centered on the static path. The initial population does not lose randomness, and the overall quality is improved, which can speed up the searching speed.

3.3. The global optimum location based on simulated annealing criterion

The initial stage of the pigeon swarm algorithm uses the map operator to search path. In each iteration, the pigeons will get a new position and speed. In the velocity iteration formula of map operator, e^{-Rt} is a decreasing exponential function, which will approach to zero in the later iteration. Therefore, the map operator is very dependent on the global optimal position x'_g . In this paper, the global optimal position is improved, and the improved geomagnetic operator update formula is as follows:

$$\begin{cases} v_i(t) = v_i(t-1)e^{-Rt} + rand \cdot (x'_g - x_i(t-1)) \\ x_i(t) = x_i(t-1) + v_i(t) \end{cases} \quad (12)$$

Where $v_i(t)$ is the speed of the i -th pigeon after the t -th iteration. $x_i(t)$ is the position of the i -th pigeon after the t -th iteration. R is the map factor. $rand$ is a random number of $[0, 1]$. x'_g is the improved global optimal position in the iteration.

In this paper, Gaussian perturbation is introduced to the original global optimal position to avoid trapping the local optimal position with the pigeons colony algorithm and improve the diversity of the entire population. The details are as follows:

$$x_G(t) = x_g(t) + r_1 G \quad (13)$$

Where $x_g(t)$ is the original global optimal position.

$x_G(t)$ is the position of the global optimal position after Gaussian perturbation. $r_1 \in [0, 1]$ is the control parameter. G is a Gaussian distribution with a mean of 0 and a variance of 1, and its probability density function is as follows:

$$G = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad x \in (-\infty, +\infty) \quad (14)$$

In addition, in the search process, the poor solution should be accepted with a certain probability, which can effectively avoid falling into local optimum in the iterative search process. In this paper, simulated annealing algorithm is introduced to solve this problem. Let the fitness value of the original global optimal position x_g before x_g perturbation be $f(x_g)$ and the fitness value of the global optimal position x_G after Gaussian perturbation be $f(x_G)$. If $f(x_G) > f(x_g)$, or the formula (13) is correct, then the disturbed position x_G is taken as the improved global optimal position x'_g . Otherwise, the original perturbation global optimal position x_g is still used as the improved global optimal position x'_g , as follows:

$$\begin{cases} \exp\left(\frac{-(f(x_g) - f(x_G))}{KT}\right) > rand \\ T(t) = \mu \times T(t-1) \end{cases} \quad (15)$$

In the formula, K is Boltzmann constant. μ is the attenuation factor. $T(t)$ is the current iteration annealing temperature, which decreases gradually with the iteration increasing.

3.4. The pigeon number with adaptive step length

In the landmark operator, the population quantity is too small in the later iteration period, which affects the optimization of the algorithm. In the stage of landmark operator, the population size can be slightly larger at the beginning of iteration [27-29], but it should gradually decrease as the iteration number increases. Logsig function has the characteristics of nonlinear reduction from 1 to 0, so Logsig function is introduced in this paper as the adaptive step of the pigeon number, and the improved landmark operator update formula is shown as follows:

$$\begin{cases} N_p(t) = N_{pmax} \times \log sig((N_{cmax2} / 2 - t) / k) \\ x_i(t) = x_i(t-1) + rand \cdot (x_c(t) - x_i(t-1)) \\ x_c(t) = \left[\sum_{i=1}^{N_p(t)} x_i(t) \cdot f(x_i(t)) \right] / \left[N_p(t) \cdot \sum_{i=1}^{N_p(t)} f(x_i(t)) \right] \end{cases} \quad (16)$$

Where $N_p(t)$ is the number of pigeons in the current iteration number. $N_{p_{max}}$ represents the maximum number of population size. $N_{c_{max2}}$ is the maximum number of iterations in the landmark operator stage. k is the slope of the logsig function. $x_c(t)$ represents the center of the t -th generation group. $f(x_i(t))$ is the fitness function of the i -th pigeon. After the improvement, the population size shows a nonlinear decreasing trend, and its value decreases gradually, which ensures the diversity of the population.

3.5. Path smoothing and replanning

In actual process, the path should be smoothed as much as possible. In this paper, Lagrange curve is used to smooth the path. A schematic diagram of smoothing curves is shown in figure 3.

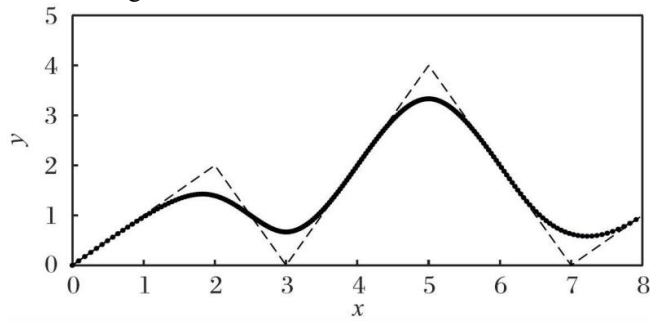


Figure 3. Soothing process. (dotted line: original curve; full line: smoothed curve)

After smoothing, if a part of the path enters the inside of the obstacle, the intersecting collision part of the path can be reprogrammed to make the path closer to the original curve [30]. The specific operation is as follows.

As shown in figure 4, when the smoothed path enters the inside of the obstacle, but the original path does not enter the inside of the obstacle, the new path points are selected from the midpoint of the three sides formed by the four points for smoothing, and the above process is repeated until the collision is eliminated.

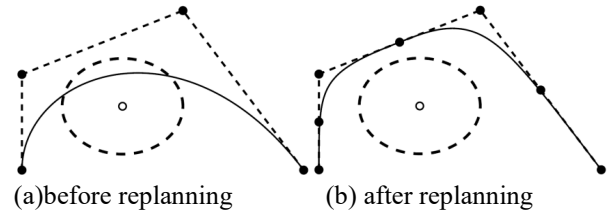


Figure 4. Replanning diagram.

According to the above improved scheme, the overall steps of this paper are as follows:

- 1) Establish the map model with the grid method, and set the parameters of the global static stage, including the ant colony number m , pheromone heuristic factor α , heuristic information factor β , pheromone volatility factor ρ , pheromone intensity coefficient Q , iteration times N_c , etc.,
- 2) Using the improved A* ant colony algorithm to plan a static offline path, and smooth the path.
- 3) The nodes of the static path are used to initialize the improved pigeon algorithm in the form of normal distribution, and the parameters of the local dynamic obstacle avoidance stage are set, including the number of pigeons $N_{p_{max}}$, the iteration number of the map and landmark operator $N_{c_{max1}}$ and $N_{c_{max2}}$, the map factor R , the initial annealing temperature $T(0)$, the attenuation factor μ , and the slope of the logsig function k .
- 4) When the mobile robot detects the dynamic obstacle by its own sensor, the previous grid point of the detected obstacle is used as the local starting point, and the improved pigeon swarm algorithm is used to carry out local dynamic obstacle avoidance, and the local path is smoothed. Otherwise, it follows the original static path to the destination point.
- 5) To judge whether the mobile robot has reached the target point. If YES, the whole path planning process will be ended. Otherwise, it returns to Step 4.

The overall process is shown in figure 5.

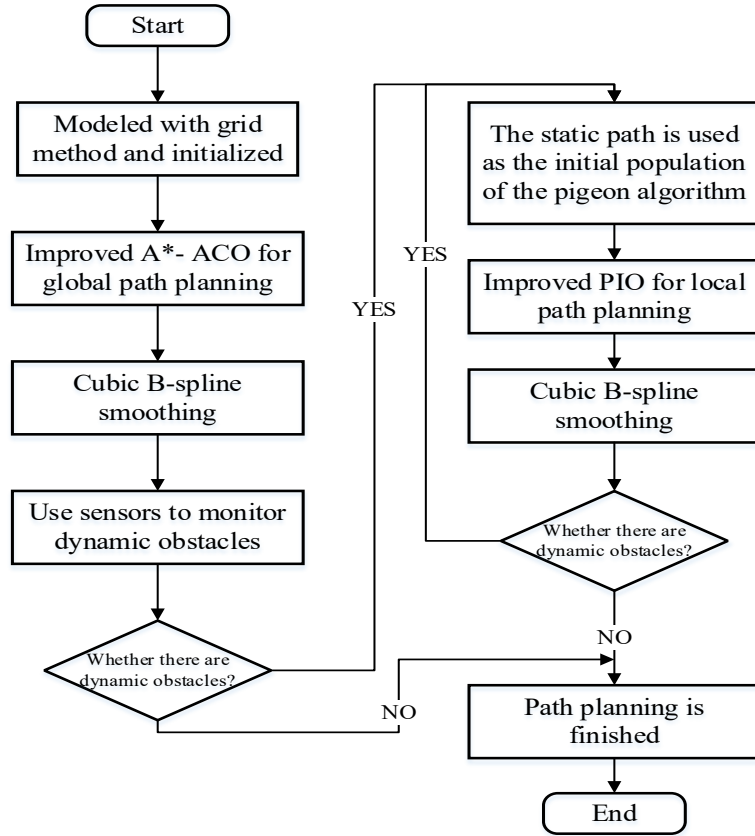


Figure 5. Proposed path planning process.

4. Experimental results and analysis

The performance of the proposed algorithm is simulated from both static and dynamic aspects.

Based on the grid method, the modeling and simulation analysis is carried out under Windows 10 system, MATLAB 2017b platform.

4.1. Global static obstacle avoidance

In order to verify the performance of the proposed algorithm in the process of global static obstacle avoidance, simulation comparisons are made under different environments, it requires that the path should not collide with the edge of the obstacle. In the global static obstacle avoidance stage, the number of ant colony $m=20$, the number of iterations $Nc=30$, the pheromone heuristic factor $\alpha =1$, the pheromone heuristic factor $\beta =12$, the pheromone volatile factor $\rho =0.2$, and the pheromone intensity coefficient $Q=10$.

Simulation comparison in environment 1.

In environment 1, the traditional ant colony algorithm and the proposed algorithm are compared and analyzed. The

starting point coordinate is (5.4,5.4) and the target point coordinate is (28.1,28.1). Figure 6(a) is the optimal path planned by the traditional ant colony algorithm, and Figure 6(b) is the optimal path planned by the proposed algorithm in this paper.

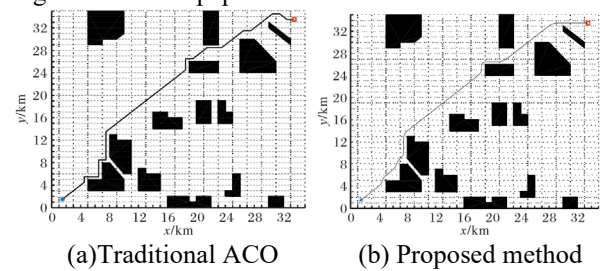


Figure 6. Comparison of traditional ant colony algorithm and the proposed algorithm under static environment 1.

Simulation comparison in environment 2.

In environment 2, the traditional ant colony algorithm and the proposed algorithm are compared and analyzed. The starting point coordinates are (0.8, 36.7) and the target point coordinates are (32.4,0.6). In here, Figure 7(a) is the

optimal path planned by the traditional ant colony algorithm, and Figure 7(b) is the optimal path planned by the proposed algorithm.

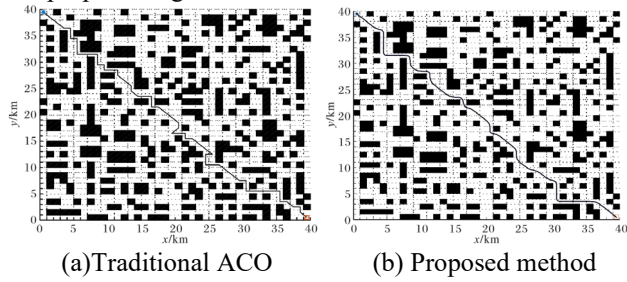


Figure 7. Comparison of traditional ant colony algorithm and the proposed algorithm under static environment 2.

In the static environment, the performance indexes of the proposed algorithm and the traditional ant colony algorithm in environment 1 and environment 2 are compared and analyzed. Table 1 and table 2 present the 10 times path planning results with the traditional ant colony algorithm and the proposed algorithm in the two environments. It can be seen that the traditional ant colony algorithm finds the optimal path 52.527 km for two times in environment 1 and 71.452 km for 3 times in environment 2. The proposed algorithm in this paper find the optimal path 49.937 km for 10 times in environment 1 and 66.280 km for 4 times in environment 2. In the two environments, the mean values of proposed algorithm are reduced by 2.977km and 5.078km than the traditional ant colony algorithm, respectively. The average execution time of the proposed algorithm in environment 1 is 2.501s, which is reduced by 7.71% than that of the traditional ant colony algorithm (2.710s). The average execution time of the proposed algorithm in environment 2 is 4.870s, which is reduced by 6.38% than that of the traditional ant colony algorithm (5.202s). Overall, the proposed algorithm in this paper has better performance.

Table 1. Performance comparison between traditional ant colony algorithm and the proposed algorithm under environment 1

Number	ACO		Proposed	
	Length/m	Time/s	Length/m	Time/s
1	53.392	2.761	49.937	2.521
2	52.827	2.575	49.937	2.470
3	52.527	2.561	49.937	2.549
4	53.392	2.702	49.937	2.464
5	52.954	2.832	49.937	2.481
6	52.643	2.881	49.937	2.483
7	52.523	2.542	49.937	2.439
8	52.921	2.820	49.937	2.557
9	53.438	2.717	49.937	2.452
10	52.527	2.689	49.937	2.530
Average	52.916	2.710	49.937	2.501

Table 2. Performance comparison between traditional ant colony algorithm and the proposed algorithm under environment 2

Number	ACO		Proposed	
	Length/m	Time/s	Length/m	Time/s
1	72.374	5.209	67.176	4.832
2	71.452	5.238	66.845	4.947
3	71.730	5.269	66.910	5.062
4	71.739	5.182	66.280	4.841
5	72.071	5.150	66.541	4.836
6	71.452	5.212	66.280	4.909
7	72.011	5.274	66.280	4.722
8	71.791	5.136	67.427	4.834
9	71.452	5.139	66.280	4.748
10	72.019	5.212	67.290	4.966
Average	71.809	5.202	66.731	4.870

4.2. Local dynamic obstacle avoidance

In order to verify the performance of the proposed algorithm in the process of local dynamic obstacle avoidance, simulation comparisons are made under different environments. The number of pigeons in the local dynamic obstacle avoidance stage is $N_{p_{max}}=150$, the iteration time is $N_{c_{max1}}=150$, $N_{c_{max2}}=100$, the initial annealing temperature is $T(0)=100$, the attenuation factor is $\mu=0.99$, and the slope of logsig function is $k=5$.

Simulation comparison in environment 1.

In environment 1, the path of the obstacle and the mobile robot partially overlaps, as shown in figure 8.

Point P in the figure is the grid point before the collision point is detected, and point M is the next grid point after the obstacle leaves the trajectory of the mobile robot. In other words, the obstacle at this time only moves between point P and point M. P and M are local starting points and local target points, and the coordinates are (9.5, 15.5) and (17.5, 23.5), respectively.

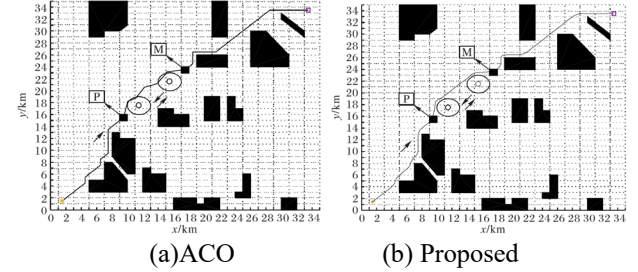


Figure 8. Comparison of traditional ant colony algorithm and the proposed algorithm under dynamic environment 1

As can be seen from Figure 8(a), the path performance obtained by the traditional pigeon swarm algorithm is poor, and the obstacle cannot be completely avoided. It will still collide with the edge of the obstacle. In figure 8(b), the path obtained by the proposed algorithm in this paper keeps a certain distance from the obstacle, and the length of the path is shorter and smoother.

Simulation comparison in environment 2.

In order to further verify the performance of the algorithm in the dynamic obstacle avoidance stage, simulation comparison is conducted in environment 2. That is, in the complex environment, where a large number of sudden dynamic obstacles are randomly generated in the planned space, as shown in Fig. 9. Point S represents the initial point of planning, point E represents the target point of planning, and the circular area represents the dynamic obstacle after expansion. As can be seen from Figure 9, the traditional pigeon algorithm cannot completely avoid obstacles when confronted with a large number of sudden obstacles. The path obtained by the proposed algorithm in this paper has a shorter path length and is smoother, which can achieve complete obstacle avoidance.

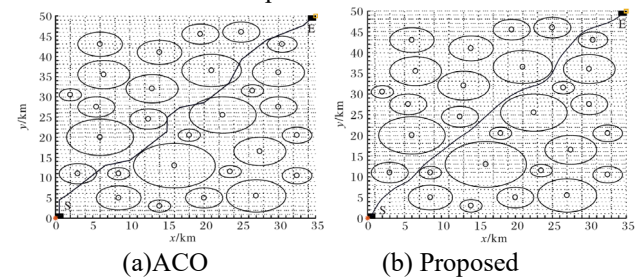


Figure 9. Comparison of traditional ant colony algorithm and the proposed algorithm under dynamic environment 2

In the dynamic environment, the performance indexes of the proposed algorithm and the traditional pigeon algorithm are compared and analyzed in environment 1 and environment 2. Table 3 and table 4 are the performance index comparison table of 10 experiments conducted in environment 1 and environment 2 between the traditional pigeon colony algorithm and the proposed algorithm. It can be seen that in environment 1, the average path length of the proposed algorithm in 10 experiments is 51.269km, it is reduced by 0.133km than that of the traditional pigeon algorithm, and the average value of the proposed algorithm is 2.619, which is reduced by 0.171 than that of the traditional pigeon colony algorithm (2.791).

Table 3. Performance comparison between traditional PIO and the proposed algorithm under environment 1

Number	PIO		Proposed	
	Length /m	Evaluation value	Length/m	Evaluation value
1	51.264	2.696	51.387	2.650
2	51.383	2.770	51.260	2.632
3	51.201	2.741	51.350	2.684
4	51.538	2.793	51.231	2.577
5	51.558	2.718	51.284	2.594
6	50.914	2.923	50.924	2.711
7	51.763	2.866	51.153	2.534
8	51.879	2.891	51.066	2.511
9	50.911	2.620	52.027	2.720
10	51.601	2.871	51.016	2.581
Average	51.401	2.791	51.269	2.619

Table 4. Performance comparison between traditional PIO and the proposed algorithm under environment 2

Number	PIO		Proposed	
	Length /m	Evaluation value	Length/m	Evaluation value
1	69.544	91.945	61.459	71.525
2	70.639	85.661	62.960	72.188
3	68.890	91.709	62.468	70.561
4	69.452	93.822	61.789	72.672
5	71.624	86.191	62.331	74.494
6	70.641	88.753	62.628	73.912
7	68.434	86.644	63.016	72.736
8	71.594	89.251	62.809	71.763
9	68.974	92.650	62.845	74.194
10	70.232	91.753	63.324	73.269
Average	70.002	89.838	62.563	72.731

It can be seen from the above results, under the same environment, the proposed algorithm in this paper can plan a path with shorter length, smaller evaluation function value and better performance, so it has stronger optimization ability.

5. Conclusion

The convergence speed of traditional ant colony algorithm is slow and it is easy to fall into the local optimal solution, so it is difficult to find the global optimal solution when applied to robot path planning. In this paper, an improved bidirectional A* algorithm is introduced to the traditional ant colony algorithm. The transition probability and pheromone updating mechanism of ant colony algorithm are improved. Compared with the traditional ant colony algorithm, the global static path algorithm in this paper can obtain a better static path. In view of the dynamic obstacles in the process of movement, this paper proposes an improved pigeon algorithm to achieve local dynamic obstacle avoidance. By improving the map and landmark operator of the traditional pigeon algorithm, the improved local dynamic obstacle avoidance algorithm can achieve better obstacle avoidance compared with the traditional

pigeon algorithm. Finally, this paper introduces cubic B-spline curve to smooth the whole path to make the path more realistic.

Acknowledgments.

This work was supported by the 1) Major Science and Technology Innovation Project of Shandong Province (No. 2019JZZY010128) and 2) Natural Science Fund Project Guidance Plan in Liaoning Province of China and Joint Fund for Robotics (No. 20180520024).

References

- [1] Kaleci, B., Eler, A., Dutaac, H., et al. (2020) Semantic classification of mobile robot locations through 2D laser scans. *Intelligent Service Robotics* 13(4).
- [2] Parra, C., Cebollada, S., Paya, L., et al. (2020) A novel method to estimate the position of a mobile robot in underfloor environments using RGB-D point clouds. *IEEE Access* 8: 9084-9101.
- [3] Liang, J. (2011) A Path Planning Algorithm of Mobile Robot in Known 3D Environment. *Procedia Engineering* 15:157-162.
- [4] Shan, B., Zhang, D. (2016) Path Planning of Robot Based on Ant Colony Optimization Algorithm. *Applied Mechanics and Materials* 614:199-202.
- [5] Pan, J., Nguyen, T., Chu, S., et al. (2018) A Multi-objective Ions Motion Optimization for Robot Path Planning[C]// *International Conference on Engineering Research and Applications*. Springer, Cham, 2018.
- [6] Collins, L., Ghassemi, P., Esfahani, E., et al. (2021) Scalable Coverage Path Planning of Multi-Robot Teams for Monitoring Non-Convex Areas. *arXiv:2103.14709*
- [7] Ferguson, D., Stentz, A. (2006) Using interpolation to improve path planning: The Field D* algorithm. *Journal of Field Robotics* 23(2):79-101.
- [8] Chaari, I., Koubaa, A., Bennaceur, H., et al. (2014) On the Adequacy of Tabu Search for Global Robot Path Planning Problem in Grid Environments. *Procedia Computer Science* 32:604-613.
- [9] Corno, F., Prinetto, P., Rebaudengo, M., et al. (2006) GATTO: a genetic algorithm for automatic test pattern generation for large synchronous sequential circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 15(8):991-1000.
- [10] Wang, W., Dong, P., Zhang, F. (2018) The shortest path planning for mobile robots using improved A~* algorithm. *Journal of Computer Applications* 38(5): 1523-1526.
- [11] Hu, C., Xia, Y., Zhang, J. (2018) Adaptive Operator Quantum-Behaved Pigeon-Inspired Optimization Algorithm with Application to UAV Path Planning. *Algorithms* 12(1).
- [12] Huang, Y., Yi, Q., Min, S. (2013) An Improved Dijkstra Shortest Path Algorithm. *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (Icsee)*, 2013.
- [13] Du, W., Cao, X., Lin, Z., et al. (2009) Evolutionary games on scale-free networks with a preferential selection mechanism. *Physica A Statistical Mechanics & Its Applications* 388(20):4509-4514.
- [14] Liu, C., Wei, X., Yang, J. (2011) A path planning method based on improved genetic algorithm for mobile robot in static environment.
- [15] Tan, G., He, H., Sloman, A. (2006) Global optimal path planning for mobile robot based on improved Dijkstra algorithm and ant system algorithm. *Journal of Central South University of Technology*, 2006(1):80-86.
- [16] Wang, Y., Yu, X., Liang, X. (2018) Design and Implementation of Global Path Planning System for Unmanned Surface Vehicle among Multiple Task Points. *International Journal of Vehicle Autonomous Systems* 14(1).
- [17] Laghari, A., He, H., Shafiq, M., et al. (2018) Assessment of quality of experience (QoE) of image compression in social cloud computing. *Multiagent and Grid Systems* 14(2): 125-143.
- [18] Shahid, K., Zhang, Y., Laghari, A., et al. (2017) Image processing based proposed drone for detecting and controlling street crimes. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pp. 1725-1730. *IEEE*, 2017.
- [19] Laghari, A., He, H., Khan, A., et al. (2018) Impact of video file format on quality of experience (QoE) of multimedia content. *3D Research* 9(3): 39.
- [20] Karim, S., Zhang, Y., Yin, S., Laghari, A., Ali, B. (2019) Impact of compressed and down-scaled training images on vehicle detection in remote sensing imagery. *Multimedia Tools and Applications* 78(22): 32565-32583.
- [21] Chen, L., Sun, H. (2021) Picking path optimization of mobile robotic arm based on differential evolution and improved A* algorithm. *IEEE Access*.
- [22] Qu, B., Xu, Y., Wang, D., et al. (2014) Path Planning Using Neighborhood Based Crowding Differential Evolution[C]// *International Conference in Swarm Intelligence*. Springer, Cham, 2014.
- [23] Biagiotti, L., Melchiorri, C. (2013) Online trajectory planning and filtering for robotic applications via B-spline

- smoothing filters[C]// IEEE/RSJ International Conference on Intelligent Robots & Systems. IEEE, 2013.
- [24] Yin, S., and Li, H. (2020) Hot Region Selection Based on Selective Search and Modified Fuzzy C-Means in Remote Sensing Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13: 5862-5871.
- [25] Liu, A., Jiang, J., Xu, K. (2020) Robot path planning based on improved ant colony and pigeon inspired optimization algorithm. *Journal of Computer Applications* 40(11): 3366-3372.
- [26] Duan, H., Qiao, P. (2014) Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. *International Journal of Intelligent Computing and Cybernetics* 7(1), pp.24-37.
- [27] Yin, S., Liu, J., Zhang, Y., et al. (2016) Cuckoo search algorithm based on mobile cloud model. *International Journal of Innovative Computing, Information and Control* 12(6): 1809-1819.
- [28] Liu, T., Yin, S. (2017) An improved particle swarm optimization algorithm used for BP neural network and multimedia course-ware evaluation[J]. *Multimedia Tools & Applications* 76(9):11961-11974.
- [29] Sun, Y., Yin, S., Li, H., et al. (2019) GPOGC: Gaussian Pigeon-Oriented Graph Clustering Algorithm for Social Networks Cluster. *IEEE Access* 7: 99254-99262.
- [30] Sun, Y., Yin, S., Teng, L. (2020) Research on Multi-robot Intelligent Fusion Technology Based on Multi-mode Deep Learning. *International Journal of Electronics and Information Engineering* 12(3): 119-127.