U.P.B. Sci. Bull., Series C, Vol. 85, Iss. 3, 2023 ISSN 2286-3540

RESEARCH ON SERVICE ROBOT PATH PLANNING BASED ON OPTIMIZED A-STAR ANT COLONY ALGORITHM

Yunfei XIA[1,*], Yang ZHENG[2], Xing WANG[3], Ping JIN[4], Liangyan WANG[5]

In the past, service robots planning paths based on the results of heuristic functions did not guarantee the search for optimal paths, and there were also problems such as more turning points in the planned paths and difficulty in effective dynamic obstacle avoidance. In this paper, a hierarchical strategy is used for regional path planning. An adaptive pheromone fluctuation factor is defined to optimize the node selection method. Add the constraint of obstacles on the speed. The improved algorithm has a wider search range and faster convergence at a later stage. The service robot can reach the designated place faster in the environment with different complexity.

Keywords: optimized A-star ant colony algorithm, hierarchical strategy, real-time dynamic obstacle avoidance

1. Introduction

The current service robot achieves path planning and autonomous obstacle avoidance in ROS environment through multi-sensor fusion, SLAM and motion control technologies. For robots driving in environments with different complexity such as parks and warehouses, this paper optimizes the paths through hierarchical strategies to achieve planning the optimal paths for multiple environments with different complexity and real-time obstacle avoidance. For global path planning, literature [1] proposed an improved ant colony algorithm, which updates the pheromone according to the sorting results, and plays a guiding role in the optimal and suboptimal paths, with high robustness, but there are problems such as large amount of computation and slow convergence speed. In [2], the ant colony algorithm was improved by using the non-uniform initial pheromone value, which reduced the blindness of initial search. The convergence speed was fast, but it was not conducive to exploration. The diversity was poor,

[1] Prof., School of Robotics Engineering Anhui Sanlian University, China, e-mail: yunfeixia163@163.com
[2] Prof., School of Robotics Engineering Anhui Sanlian University, China, e-mail: zy85618683@163.com
[3] Prof., School of Robotics Engineering Anhui Sanlian University, China, e-mail: jqrxy-ky@slu.edu.cn
[4] Prof., School of Robotics Engineering Anhui Sanlian University, China, e-mail: sirenrabbit@sina.com
[5] Prof., School of Robotics Engineering Anhui Sanlian University, China, e-mail: wang_ly2442@163.com

and it was easy to fall into local optimization. In [3], the triangulation algorithm was used to reduce the redundant nodes of the A-star algorithm, but the smoothness of the path was not considered. In [4] and [5], the premature and slow convergence problems of the ant colony algorithm were optimized, but the algorithm took a long time. A multi-objective travel planning algorithm based on ant colony was proposed in [6], but this algorithm lacks dynamic obstacle avoidance characteristics. The QAPF learning algorithm was proposed in [7, 8] to calculate the optimal path of a robot, but its convergence speed to the optimal solution is slow. An optimized artificial potential field method was proposed in [9], which is robust to control and sensing errors, but has the disadvantage of having local minima. Another method is to enhance the ant colony genetic algorithm by adaptively using advanced solution pheromones, but this algorithm is highly dependent on the advantages of the initial population, as shown in [10]. Therefore, in this paper, the region is partitioned according to the complexity of the environment. The high-density environment is used by adaptive ant colony algorithm, and path planning is performed in the low-density environment with multi-objective point A-star algorithm to improve path planning efficiency and reduce redundant turning points. The dynamic obstacle motion model is established, and the robot motion behavior constraints are set to improve the robustness of obstacle avoidance in dynamic environments, so that the planned paths are more consistent with the kinematic laws of service robots and have good obstacle avoidance functions in dynamic environments.

2. Optimized Ant Colony Algorithm

There are two main key steps in the ant colony algorithm, namely path construction and pheromone update. Path construction is completed by storing the path points that ants pass through in each iteration. Ants will store the pheromone of their path and choose the planning path according to the pheromone concentration and heuristic factor, as shown in equation (1).

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum_{s \in J_k(i)} [\tau_{is}(t)]^{\alpha} \cdot [\eta_{is}(t)]^{\beta}} \tag{1}$$

where τ is the pheromone concentration of the selected path, and η is the heuristic factor associated with the length of the selected path. α and β are used to indicate the pheromone concentration and the importance of the heuristic factor.

All ants come back to complete one iteration, and after each iteration, the pheromone of all the routes taken will be updated, when t+1 iterations pheromone concentration is updated as shown in equation (2).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \tag{2}$$

Δτ is the pheromone increment of the ant during this iteration, expressed by the pheromone intensity Q and the total length L of the path taken by the Kth ant in this iteration, as shown in (3).

$$\Delta\tau_{ij}(t) = \frac{Q}{L_k} \tag{3}$$

As in the literature [11], [12], the pheromone update rule is improved and a reward and punishment mechanism is proposed to enhance the pheromone concentration of optimal and better paths after each iteration, while weakening the pheromone concentration of inferior paths to enhance the guiding effect on superior paths, as shown in equation (4).

$$\Delta\tau_{ij}(T) = \frac{\lambda(X-N)}{2} \times \frac{T+1}{T_{max}} \times \frac{Q}{L_k} \tag{4}$$

T represents the current iteration number, N denotes the Nth path selected at the Tth iteration number when the paths are sorted from best to worst, and X denotes the total number of ants. λ is a constant greater than zero.

As in the literature [13] for the pheromone volatility factor ρ(t), the pheromone volatility factor as an important factor for pheromone update, this paper uses the adaptive pheromone volatility strategy, this improves the search range of the algorithm at the initial iteration stage and reduces the time required for later convergence. The pheromone volatility coefficient, which decreases adaptively with the number of iterations, does not decrease after decreasing to a suitable range, as shown in equation (5).

$$\rho(t) = Ce^{-\left(\frac{t}{2\sigma^2}\right)^2} \tag{5}$$

Where C is the scale factor, optionally 0.5 after experimental validation, associated with the peak of the normal distribution, t is the number of iterations, and σ is the retardation of the normal distribution, adjustable according to the expected rate of convergence, optionally 72.

3. Optimized A-star Algorithm

The A-star algorithm first needs to rasterize the map and search the surrounding eight rasters separately from the starting point, by calculating the minimum total generation value of the surrounding eight rasters as the starting point for the next calculation. The cycle continues until the end of the surrounding raster appears. The equation for the total generation value is shown in (6).

$$F^*(i) = G(i) + H^*(i) \tag{6}$$

For the A-star algorithm, the common Euclidean evaluation function is shown in equation (7).

$$\boldsymbol{H(i) = d \times \sqrt{(i_x - g_x)^2 + (i_y - g_y)^2}} \tag{7}$$

Although the A-star algorithm plans paths in the raster map, it can only satisfy the connections between adjacent rasters. Here, the global path optimization function is achieved by removing redundant nodes and fitting the remaining nodes. This method first needs to traverse the original A-star algorithm. Except for each node of the start and target nodes, when the adjacent nodes before and after the node are connected without touching obstacles, delete the node. Loop this method and connect the remaining nodes to form a shorter path to reach the target point, as shown in Fig. 1.

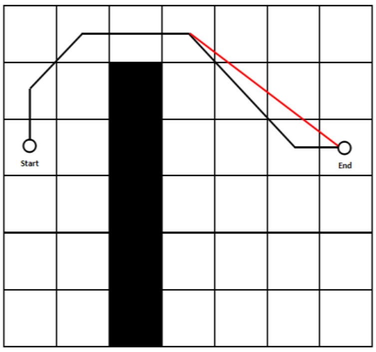

Fig. 1. Delete redundant node path planning

4. Defining Dynamic Obstacle Motion Models

First the service robot will build a speed model based on its own hardware. Second, the robot motion trajectory is simulated in a short time slice based on the sampled velocity model. Finally, the evaluation function of the robot moving trajectory is set and the velocity model of the optimal path is selected. The evaluation function established by the azimuth angle, velocity model and obstacle distance length is shown in equation (8).

$$E(v,w) = \theta[\alpha \cdot deviation(v,w) + \beta \cdot obstacle(v,w) + \gamma \cdot speed(v,w)] \quad (8)$$

The deviation evaluation function represents the deviation degree of the simulated trajectory. The obstacle distance evaluation function represents the distance between the current track and the obstacle, and the speed evaluation function represents the appropriateness of the speed.

The dynamic windowing method has poor robustness and cannot achieve timely obstacle avoidance, as described in the literature [14,15]. These problems are solved here by setting up a dynamic obstacle motion model to predict the range of obstacle movement. When dynamic obstacles are encountered, the dynamic obstacle motion model needs to be calculated, and the speed of the dynamic obstacle remains constant in a shorter time slice, then the speed model of the obstacle is obtained according to the sensor. According to the velocity model, the range of obstacles in a time period is calculated, and the service robot needs to perform dynamic obstacle avoidance according to this obstacle range, and the dynamic obstacle motion model is shown in equation (6).

$$\begin{cases} x = x + v\Delta t\, cos(\theta_t) - v_y \Delta t\, sin(\theta_t) \\ y = y + v\Delta t\, sin(\theta_t) - v_y \Delta t\, cos(\theta_t) \\ \qquad \theta_t = \theta_t + w\Delta t \end{cases} \tag{9}$$

5. Optimized A- star Ant Colony Hybrid Algorithm

For a map of the same complexity, the corresponding path planning algorithm can play the corresponding advantage, if for a map with different complexity, it is necessary to improve the reliability and operational efficiency of the planning based on the map complexity metric and the corresponding algorithm for planning. Currently, for raster maps, a map complexity measure based on the relative Hemming distance is used to partition the map area.

Based on environments with different complexity, such as robots working in and out of warehouses for logistics. If path planning is performed by the same global path planning algorithm, it leads to getting into deadlocked regions with unreliable path planning in regions of higher complexity or increases computation and wastes computation time in regions of lower complexity. To solve this problem, a hierarchical strategy for path planning based on the map complexity metric is designed here. The strategy first requires rasterizing the map, then dividing the regions by the complexity measure, and finally using appropriate path planning algorithms according to different complexity regions.

The raster map is represented as a two-dimensional matrix of 0,1 with obstacle points as 1 and no obstacle points as 0. The complexity of the map is displayed by comparing the difference in the corresponding bit positions of the rows and columns in the matrix. Because the number of bits is the same, comparing the bits of two adjacent columns, the more bits are different, the greater the complexity. yy indicates the number of corresponding bits in two columns that are 1 at the same time, yn indicates the number of corresponding bits that are different, and nn indicates the number of corresponding bits that are 0 at the same time. The Hemming function is shown in equation (10).

$$HammingDist(A,\ B) = YN + NY \tag{10}$$

The raster map is represented as a matrix M, M=(a1,a2,a3,...,am), and m is the length of the horizontal axis X of the map, so the X-axis directional complexity metric of a raster map, as shown in equation (11) ,the Y-axis direction is shown in equation (12).

$$x\_HC(M) = \sum_{i=1}^{m} HammingDist(a_i, a_{i-1}) \tag{11}$$

$$y\_HC(M) = x\_HC(M^T) \tag{12}$$

When the map size is inconsistent, back affects the inconsistent results of the Hemming distance complexity measure, in order to solve this problem, it is necessary to pass the relative Hemming distance complexity measure, M is the length of the horizontal axis X in the map, n is the length of the vertical axis Y, then the matrix is M=(a1,a2,a3,...,am), then the maximum Hemming distance between map columns is n(m-1), between rows The maximum Hemming distance is m(n-1). As shown in equation (13), the Y-axis direction is shown in equation (14).

$$x\_RHC(M) = \frac{1}{n*(m-1)} \sum_{i=1}^{m-1} HammingDist\,(a_i, a_{i+1}) \tag{13}$$

$$y\_RHC(M) = \frac{n*(m-1)}{m*(n-1)}\ x\_RHC(M^T) \tag{14}$$

In practical applications, the robot moves both laterally and vertically, so the overall relative Hemming distance complexity of the map is expressed by obtaining the arithmetic mean of the relative Hemming distance complexity in the X and Y directions. As shown in equation (15).

$$RHC(M) = \frac{x\_HC(M)+y\_HC(M)}{2} \tag{15}$$

To optimize the A-star ant colony algorithm, firstly, it is necessary to carry out area complexity grading by relative Hemming distance complexity metric to divide the obstacle high-density area and low-density area. Secondly, the path planning is carried out in the high-density region by optimizing the ant colony algorithm to increase the global search range, improve the convergence speed, and avoid the service robot from entering the deadlock region. In addition, as in the literature [16], the optimized A-star algorithm is used for path planning in low-density regions. Finally, the speed and stability of dynamic obstacle avoidance are improved by the optimized DWA algorithm and obstacle avoidance based on the motion model of the set obstacles.

6. Experimental Analysis

The motion model is set according to the specific performance of the ROS robot, and the main motion parameters modeled during the robot motion are shown in Table 1.

Table 1

Robot motion model

Maximum	Maximum Rotation Speed(r/s)	Acceleration (m/s2)	Rotational Acceleration (r/ s2)	Velocity Resolution (m/s)	Speed Resolution (r/s)
1.0	80.0	0.2	20	0.01	1

Using elitist ant colony system, adaptive ant colony algorithm, and our algorithm for path planning simulation. Compared with the elitist ant colony system, our algorithm reduces the grid length by 6.6 units and 5 turning points.

Compared with the adaptive ant colony algorithm, ours reduces the grid length by 5.4 units and 2 turning points. The optimized A-star ant colony algorithm has a shorter length and smoother path, as shown in Fig. 4.

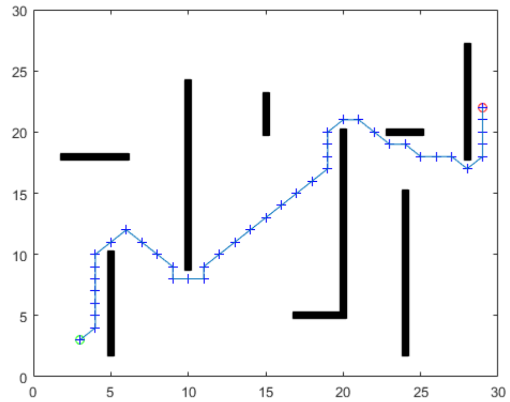

Fig. 2. Path planning of Elitist Ant Colony System

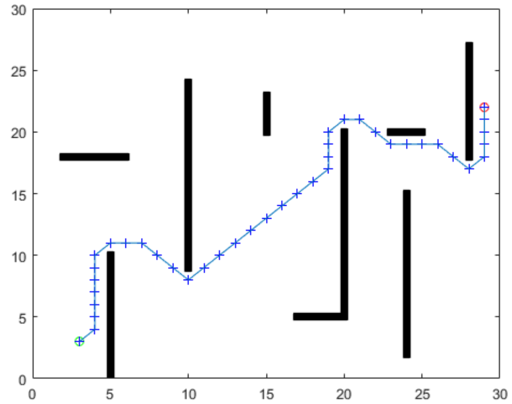

Fig. 3. Path planning of Adaptive Ant Colony Algorithm

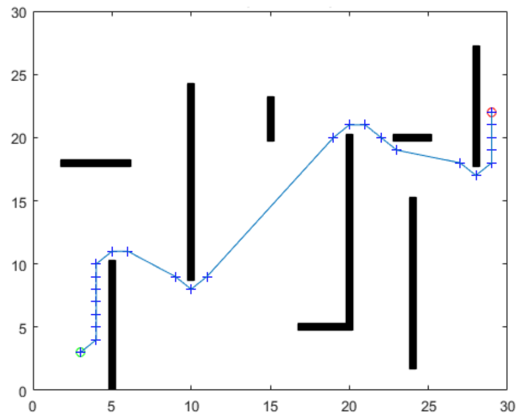

Fig. 4. Path planning of Optimized A-star Ant Colony Algorithm

Comparing the changes before and after optimization, it can be seen that the optimized algorithm has better global search ability. At the same time, it also has good iterative convergence speed. Although our algorithm is not the fastest in searching for the shortest path, it achieves faster convergence stability and shorter iterations. Meet the expected convergence requirements, as shown in Fig. 7.

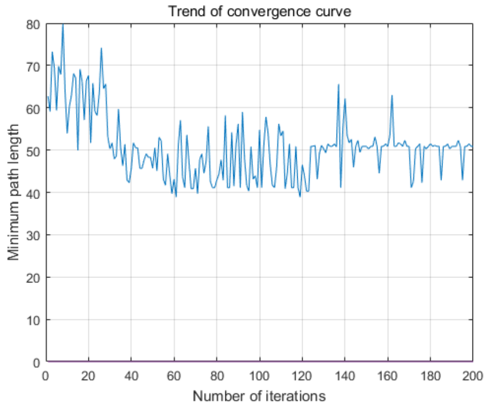

Fig. 5. Elitist Ant Colony System convergence curve

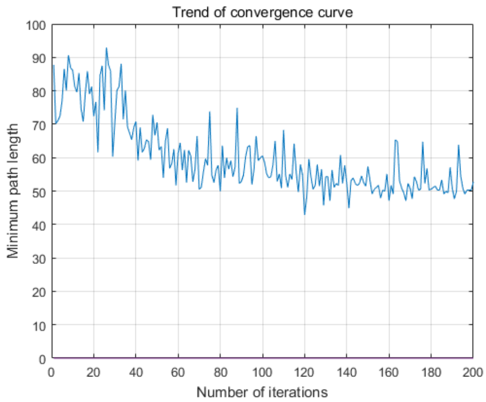

Fig. 6. Adaptive Ant Colony Algorithm convergence curve

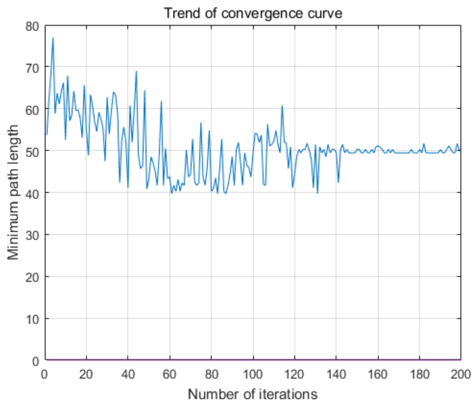

Fig. 7. Optimized A-star Ant Colony Algorithm convergence curve

Fig. 8 validates the optimised ant colony algorithm in a raster grid. This ant colony algorithm identifies multiple key target points in the global path that are typically identified at the edges of obstacles. The optimised DWA algorithm uses multi-sensor techniques for path planning of key target points and avoids obstacle points that are different from the static environment. The optimised ant colony algorithm plans the motion trajectory and the optimised DWA algorithm performs dynamic obstacle avoidance with home accuracy. Finally, the paths are smoother and do not get stuck in a deadlocked state compared to the single algorithm of the past.

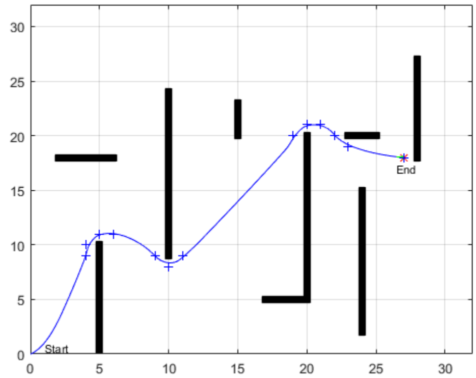

Fig. 8. Integration of dynamic path planning

Compared to the usual choice of waiting when encountering dynamic obstacles in the past, ours uses its own speed and the speed of the dynamic obstacle to determine whether to maintain speed, slow down slightly, or wait. When obstacles moving in the environment are detected, the robot slows down slightly and sets a dynamic obstacle motion model through radar and ultrasonic sensors. Traditional service robots use deceleration to avoid dynamic obstacles, which often leads to path planning errors and wastes a lot of time, as shown in Fig. 9. We validated the optimized ant colony algorithm combined with DWA. The service robot detects the security of the path ahead in real-time through its own speed model. And based on the robot's motion model and dynamic obstacles, determine whether the robot's motion strategy is to maintain normal speed or slow down and wait. Here, the robot determines that if it maintains normal speed, the two dynamic obstacles ahead will not collide with it, so it will maintain normal speed through the dynamic obstacles instead of choosing to slow down and wait, as shown in Fig. 10. Compared with previous dynamic obstacle avoidance algorithms, our algorithm has smoother speed changes and can reach the target point faster. The speed comparison is shown in Fig. 11.

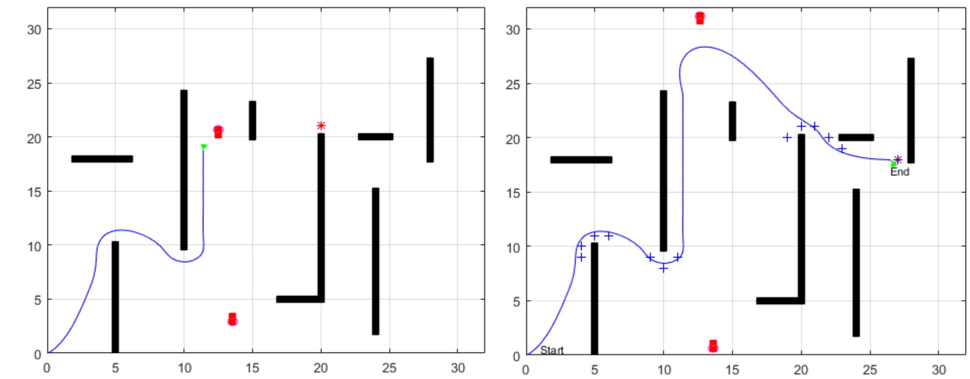

Fig. 9. Compare ant colony algorithm planning path before and after optimization

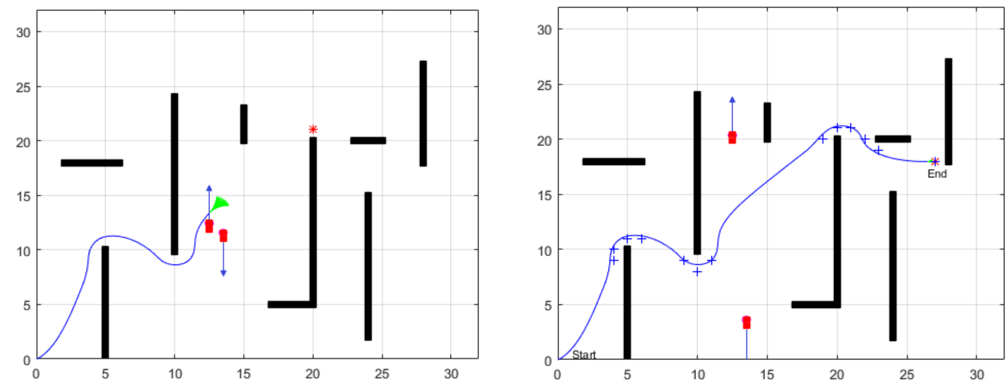

Fig. 10. Verification of dynamic obstacle avoidance strategy

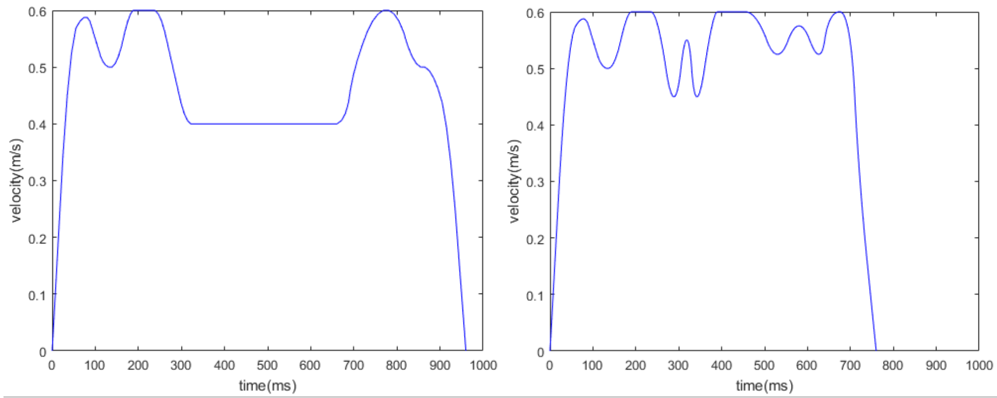

Fig. 11. Comparison of dynamic obstacle avoidance speed changes

The service robot detects obstacles in front of it at each moment and compares the speed change of obstacles by radar and ultrasonic sensors, the detection area is a sector in front of the path, and when a moving obstacle is detected, the robot decelerates slightly waiting for the action policy to be issued. Here the detected speed of the moving obstacle has no effect on the robot, then the robot will keep moving at normal speed and avoid the obstacle. The robot decelerates every time it encounters the corner of a static obstacle and a dynamic obstacle, and the speed change is shown in Fig. 12.

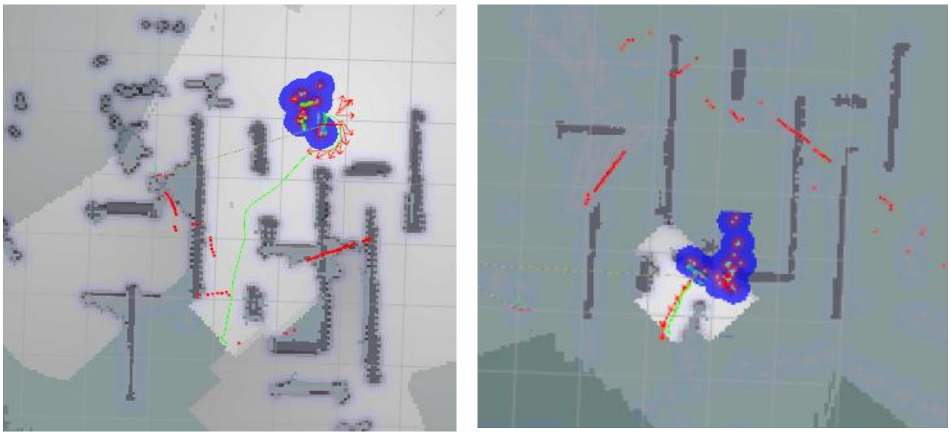

Fig. 12. ROS environment simulation

In the past, the same path planning strategy was used for areas with different complexities, using an elite ant colony system for path planning. There are more transformations, longer paths, and more iterations, as shown in Fig. 13. Although the number of iterations using adaptive ant colony algorithm is relatively reduced, it still requires a lot of computational time for areas with lower complexity, as shown in Fig. 14. We divide the global map into high complexity and low complexity regions based on the relative Hemming distance map complexity measure and mark the high complexity warehouse area with boxes. In this paper, we use the optimized ant colony DWA algorithm for path planning, which has better global search ability and faster and stable dynamic obstacle avoidance function.

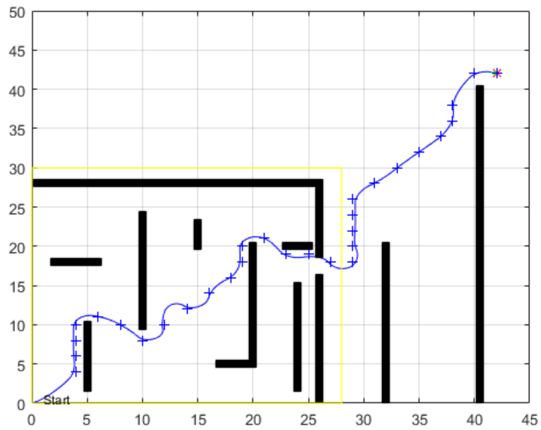

Fig. 13. Path planning of Elitist Ant Colony Systems

The service robot leaves the warehouse and uses an optimized A-star algorithm for global path planning in areas with low complexity, saving computational time and enabling real-time dynamic obstacle avoidance. Smooth path planning and more stable operation. The optimized A-star ant colony algorithm simulation is shown in Fig. 15.

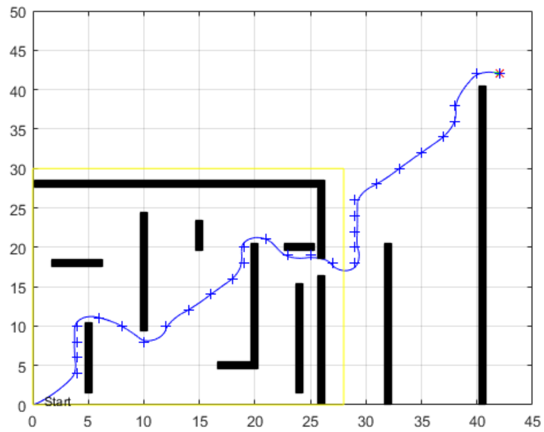

Fig. 14. Path planning of Adaptive Ant Colony algorithm

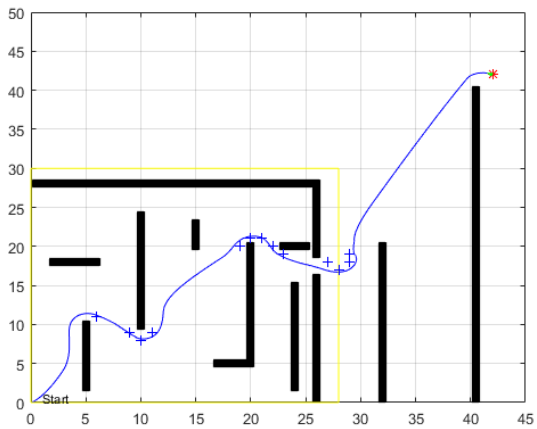

Fig. 15. Path planning of A-star Ant Colony Algorithm

This article conducted 30 experiments in the same environment and recorded the optimal path length, average path length, average number of iterations, average path planning time, and number of corners, as shown in Table 2.

Table 2

Experimental results of four indicators

parameter	Optimal path length	Average path length	Average number of iterations	Average number of turns
Elitist Ant System（EAS）	76.3	79.7	260	24
Adaptive Ant Colony Algorithm（AACO）	72.2	75.6	225	20
Optimized A-star Ant Colony Algorithm	71.6	74.5	150	16

7. Conclusion

Past A-star ant colony algorithms were slow to converge, lacked global search capability in the pre-iteration phase, and could not perform dynamic obstacle avoidance. In addition, the previous separate DWA algorithm cannot achieve optimal path planning and reliable dynamic obstacle avoidance. Therefore, an optimized A-star ant colony algorithm based on the environmental complexity index is proposed, which plans the path of the environment with different complexity levels and adds the dynamic obstacle motion model prediction to achieve a more effective obstacle avoidance strategy. The algorithm firstly divides regions by environmental complexity metric and combines the optimized A-star ant colony algorithm and the optimized DWA algorithm, which reduces the turning points in global path planning, shortens the running time, and can effectively perform dynamic obstacle avoidance. In this paper, the algorithms before and after optimization are compared and analyzed by simulating environments with different complexity and dynamic obstacles. Compared with the previous A-star ant colony algorithm, this algorithm has wider global path search range, faster convergence speed, shorter total path planning length, and is suitable for environments of different complexity. In the dynamic obstacle environment, the path is more stable and has faster and more reliable obstacle avoidance ability.

R E F E R E N C E S

[1]. Abdelbar A. M., Salama K. M. Parameter self-adaptation in an ant colony algorithm for continuous optimization. IEEE Access, 2019, 7: 18464-18479.

[2]. Mavrovouniotis M., Yang S., Van M., et al. Ant colony optimization algorithms for dynamic optimization: A case study of the dynamic travelling salesperson problem [research frontier]. IEEE Computational Intelligence Magazine, 2020, 15(1): 52-63.

[3]. Liu Z., Liu H., Lu Z., et al. A dynamic fusion pathfinding algorithm using delaunay triangulation and improved a-star for mobile robots. IEEE Access, 2021, 9: 20602-20621.

[4]. Hewawasam H. S., Ibrahim M. Y., Appuhamillage G. K. Past, present and future of path-planning algorithms for mobile robot navigation in dynamic environments. IEEE Open Journal of the Industrial Electronics Society, 2022, 3: 353-365.

[5]. Yuan J., Wang H., Lin C., et al. A novel GRU-RNN network model for dynamic path planning of mobile robot. IEEE Access, 2019, 7: 15140-15151.

[6]. Saeki E., Bao S., Takayama T., et al. Multi-Objective Trip Planning Based on Ant Colony Optimization Utilizing Trip Records. IEEE Access, 2022.

[7]. Orozco-Rosas U., Picos K., Pantrigo J. J., et al. Mobile robot path planning using a QAPF learning algorithm for known and unknown environments. IEEE Access, 2022, 10: 84648-84663.

[8]. Ayawli B. B. K., Mei X., Shen M., et al. Mobile robot path planning in dynamic environment using Voronoi diagram and computation geometry technique. Ieee Access, 2019, 7: 86026-86040.

[9]. Lazarowska A. Comparison of discrete artificial potential field algorithm and wave-front algorithm for autonomous ship trajectory planning. IEEE Access, 2020, 8: 221013-221026.

[10]. Zulfa M. I., Hartanto R., Permanasari A. E., et al. Improving Cached Data Offloading Optimization Based on Enhanced Hybrid Ant Colony Genetic Algorithm. IEEE Access, 2022, 10: 84558-84568.

[11]. Yang H., Qi J., Miao Y., et al.A new robot navigation algo-rithm based on a double-layer ant algorithm and trajec-tory optimization. IEEE Transactions on Industrial Elec-tronics, 2018.

[12]. Luo Q., Wang H., Zheng Y., et al. Research on path plan-ning of mobile robot based on improved Ant Colony algorithm. Neural Computing and Applications, 2019(1): 1-12.

[13]. Song Y., Zhang H., Cheng Chao. Path planning of logistics robot based on improved Ant Colony algorithm. Modern Manufacturing Engineering, 2022(11): 35-40+47. DOI: 10.16731/j.cnki.1671-3133.2022.11.006.

[14]. Hua Hong, Zhang Zhi'an, Shi Zhenjun, Chen Guanxing. A robot path planning method with multiple A-star algorithms in dynamic environments. Computer Engineering and Applications, 2021, 57(10): 173-180.

[15]. Zhang Yu, Song Jingzhou, Zhang Qiqi. Local path planning for outdoor sweeping robots based on improved dynamic window method. Robotics, 2020, 42(05): 617-625.

[16]. Schmid L., Pantic M., Khanna R., et al. An efficient sampling-based method for online informative path planning in unknown environments. IEEE Robotics and Automation Letters, 2020, 5(2): 1500-1507.