Optimize 2-D Path Planning of Mobile Robot by using ACO with Guidance Factor

Li Wang[1]

(Received December 12, 2019; Revised February 13, 2020; Accepted March 19, 2020)
Online First Publication
DOI: 10.33317/ssurj.148

Abstract— **In this research article, the path planning of a mobile robot is done by using Ant Colony Optimization (ACO) with guidance factor. A two-dimensional (2-D) threat map structure design for path planning strategy is utilized, in which threat points are fixed in the path of moving robot. The two main objectives of this research are; firstly, to reach the mobile robot at the target position by using optimal route using ACO strategy and secondly, by using guidance factor all the ants of ACO arrive at the fixed targeted area. Moreover, the results of the proposed algorithm compares with the classical ant system methodology. The simulated results show that the design method has short path planning and less steady state error to reach the designated path robustly.**

Index Terms— **Ant Colony Optimization (ACO), Path Planning, Mobile Robot, Guidance Factor, VORONOI Diagram.**

I. INTRODUCTION

The mobile robot track or path planning is the most basic and the most important part of mission planning. Proper trajectory planning will help robot effectively to avoid threats, shorten the distance length, and increase their survival probability and efficiency. The robot path-planning problem is a combinatorial optimization problem and an important branch of the optimization field. It is mainly through the study of mathematical methods to find the optimal arrangement, grouping or screening of discrete events. Such problems usually increase with the scale [1], [2].

The distance and time complexity for solving the problem increases exponentially, which cannot be solved by conventional methods. The path planning algorithms is divided into two categories:

- One is the traditional classic algorithm
- The other is the modern intelligent algorithm

Among them, the former mainly includes dynamic programming method, steepest descent method and optimal control method. Previously different searching methods are done which includes grid search method, artificial potential field method, neural network method, and fuzzy logic-based path planning algorithm, etc. [3].

Recently, the most commonly used method for path planning is to use the VORONOI diagram to construct the initial optional path set or set the navigation nodes. After that, select the appropriate path through the intelligent optimization algorithm. The disadvantage of this method is that the determination of the location and number of navigation nodes often requires repeated consideration. The construction of the VORONOI map determines the accuracy of the trajectory cost because the ant can only find the trajectory on the VORONOI map, not on the outer space. In addition, whenever the threat field model changes the navigation nodes and VORONOI graphs need to be reconstructed [4]. Therefore, this method is not adaptable for sudden new threats. In this paper, an Ant Colony Optimization (ACO) algorithm that introduces guidance factor is studied. Without the need to set navigation nodes and construct a VORONOI map it can automatically search for the minimum cost track in free space and has strong adaptive ability.

II. PROBLEM DESCRIPTION OF PATH PLANNING

A. *Representation of Planning Space*

This article assumes that the mobile robot keeps its speed unchanged during the whole mission and the enemy's defense zone is in a flat area. Therefore, it does not need to consider the use of terrain factors for threat avoidance maneuvers and the path-planning problem can reduce to a two-dimensional navigation trace planning problem. However, it still needs to consider the survivability of the mobile robot during the execution of the mission and the effectiveness of the mission and consider the real-time nature of the planning algorithm, so it is still a relatively special optimization problem [5]. Although the right angle grid division of planning space, from the current node to the next adjacent node, until the target node is found, the track connecting the start node and the target node is formed and the cost model and optimization algorithm based on the grid graph are used to solve the optimal path. Each node in the grid graph needs to connect the adjacent nodes with the weighted directed edge. Therefore, the data structure of the algorithm with the current node in the center. The grid size 'G' needs to be set reasonably according to the actual problem scale and the distribution of threat points.

B. *Optimized Path*

In practice, the path with acceptable range and less than a certain detectability index often use as the mission route, so the cost function shown below in the formula, i.e., Eq.1, is used to describe the performance index of the selected path, which is calculated by the weighting method of the shortest route and the least detectability path.

$$\omega = \int_0^1 \left[\beta\omega_t(S) + (1-\beta)\omega_f(S)\right] ds \tag{1}$$

Here in the formula:

[1]School of Electronic Engineering , Xi`an Aeronautical University, Xi`an , Shaanxi China.710077. Wangli871016@163.com

ω = The optimization objective function
l = The track path;
$\omega_t(S)$ = The threat cost of the route
$\omega_f(S)$ = The battery consumption cost of the route
β = The coefficient
The β indicates the intentional choice made by the route setter in the course of making the route according to the task arrangement [6].
The battery consumption cost is a function of range and the threat cost is associated with the detectability index of the mobile robot, which can be calculated on the detection probability of the mobile robot.

III. CONSTRUCTION OF ACO ALGORITHM FOR PATH PLANNING

The ACO algorithm was first used to solve the Traveling Salesman Problem (TSP). There are some difficulties in applying the algorithm to the trajectory or path planning problem, which mainly reflected in the following two aspects:

- The location and number of track nodes are not fixed. In general combinatorial optimization problems, such as TSP, the nodes of the path fixed, so it is only a combination of several nodes to achieve optimization and all nodes must go through and can only go through once. Therefore, by constructing a set of Tabu nodes (nodes that have been passed), the remaining nodes are a set of nodes to be selected, while the path planning problem is a free search space and there is no fixed node location. There is no fixed number of nodes and there are no known path nodes in the space. Therefore, these factors bring great difficulty to the planning.
- How to ensure that the target node is reached. The TSP problem returns all nodes to the starting node after one pass, so the target node is the starting node and the target node of the track-planning problem is different from the starting node. Because ACO constructs a state transition strategy according to pheromones and heuristic factors and carries out local search according to probability, the visibility is limited to the local range, so how to ensure that the target node found is the key problem to complete the track planning [7].

This research solves the above two problems by constructing ACO algorithm reasonably.
The Eq. 1, set the maximum allowable number of track nodes. The robot has a maximum range parameter, i.e., the distance of the mobile robot during the entire route that is limited by the battery and duration time allocation of the route. Keeping the maximum track length as *Lmax*, the distance *Lp* of each span should satisfy by:

$$Lp \leq Lmax \tag{2}$$

According to Eq. 2, the number of track nodes should not exceed from a certain range, which limits by the conditions of the mobile robot itself. In addition, from Eq. 1, it must be known that the path cost includes the cost of battery consumption and the track with high battery consumption may increase the track cost. The cost of battery consumption must also meet a certain range, beyond which it consider unacceptable. Therefore, although the track node is not fixed, the maximum number of nodes can be set. When the maximum number of nodes exceeds, the track is considered as infeasible and abandoned.
Taking node j as an example, if the distance between node j and target node D is djD, then the guidance factor is δj=1/djD. In the conventional state transition strategy, there is little difference in the heuristic factors of the adjacent nodes in the house map, so the local predictability of the target node is not strong, especially at the beginning of the iteration of the algorithm, i.e., the pheromones between the nodes are almost the same. The state transition is easy to fall into blind selection, so it is difficult to reach the target node quickly. According to δj=1/djD, the farther away the node is from the target node, the larger the guidance factor is; otherwise, the smaller the guidance factor is. Therefore, in the state transition strategy, the introduction of guidance factor can effectively reduce the blindness of ant search and make ants search track in the direction of the target node [8].

A. *Pheromone Up gradation*

The planning space determined according to the size of the grid divided and the positions of the starting point, i.e., the target point and the threat point. Let the coordinates of the two points in the space be (xmin, ymin) and (xmax, ymax) and set the grid size to G, then the grid has a total number of rows hN = (ymax-ymin)/G, and the total number of columns vN = (ymax-ymin)/G, where n is the number of nodes, n = hN vN, and the grid node number pj with coordinates (xj, yj) is calculated as follows in Eq. 3:

$$p_j = (\frac{x_j - x_{\min}}{G} + 1) + (\frac{y_j - y_{\min}}{G})v_N \tag{3}$$

The pheromone matrix τ establish by Eq. 3, where τ is an $n \times n$ matrix.

B. *Heuristic Factor*

The heuristic factor is mainly used to improve the visibility of node selection, speed up the convergence speed of the algorithm and make the algorithm quickly converge to the track with the minimum cost. The cost of this study mainly comes from the threat cost, so the heuristic factor is designed as the reciprocal of the threat cost point [9]. Suppose there are B threat points and the coordinate of the *i* threat point is *(xi, yi)*, then the threat cost from node *j* to threat point *i* is expressed as:

$$\varepsilon_{ji} = 1/[(x_j - x_i)^2 + (y_j - y_i)^2]^2 \tag{4}$$

According to Eq. 4, the cost of node *j* to all threat points is shown in Eq. 5.

$$\varepsilon_j = \sum_{i=1}^{B} \varepsilon_{ji} = \sum_{i=1}^{B} 1/[(x_j - x_i)^2 + (y_j - y_i)^2]^2 \tag{5}$$

The heuristic factor of a node is equal to the inverse of the total threat cost, that is $\eta_j = 1/\varepsilon_j$, so when the node threat cost is small, the heuristic factor is large and the visibility is high;

otherwise the visibility is low. The heuristic factor plays a role in accelerating the convergence of the algorithm but compared to the pheromone, the proportion should not be too large; otherwise the pheromone cannot play a guiding role [10].

C. State Transition Strategy

Let the size of the ant colony be *m* and $\tau_{ji}(n)$ represents the pheromone concentration of the track at nodes *j* to *i* when iterating n times. In the initial iteration, the pheromone concentration on each path is same. Let $\tau_{ji}(0) = h$ (*h* is a constant). Ant *k* ($k = 1, 2, \dots, m$) determines the transfer direction according to the pheromone concentration on each path during the movement. $P^k_{ji}(n)$ represents the probability of the *nth* iteration ant *k* transferring from node *j* to node *i*. The calculation formula is as follows, i.e., Eq. 6;

$$P^k_{ji}(n) = \begin{cases} \dfrac{\tau_{ji}(n)^{\alpha}\eta_{ji}(n)^{\beta}\lambda_{ji}(n)^{\gamma}}{\sum_{s\in A_k}\tau_{js}(n)^{\alpha}\eta_{js}(n)^{\beta}\lambda_{js}(n)^{\gamma}} & i \in A_k \\ 0 & other \end{cases} \tag{6}$$

$$A_k = B_i \cap \overline{T_k}$$

Among them, T_k use to record the nodes passed by ant *k* in this iteration and T_k is dynamically adjusted as ants continue to choose the next node.

Here:
T_k represents all the unpassed nodes,
B_j represents the adjacent nodes of node *j*,
A_k represents the set of nodes to be selected by ant *k* in the next step.

Unlike the general TSP problem the nodes to be selected are not all the remaining unpassed nodes T_k but a node set composed of adjacent nodes and exclude the nodes that have been passed. Ants continue to search for nodes in the local area and finally reach the target node to complete a track. In order to ensure that the ant can finally reach the target node, unlike the general ACO algorithm, the state transition probability of this study introduces the guidance factor $\delta_j(n)$, so that the ant search has a certain direction, even if the ant searches for the track in the direction of the target node, in which α, β and γ represent the importance parameters of pheromone, heuristic factor and guidance factor respectively. The simulation practice shows that the values of β and γ should not be too large; otherwise the algorithm is easy to stagnate. In addition, for the nodes in A_k that are too close to the threat point, the purpose of not selecting the node is achieved by setting the critical value *R* of the heuristic factor. When the heuristic factor η_j of a node is less than *R*, η_j is infinitesimal. Then the probability of selecting the infinitesimal node is almost zero, so as to achieve the purpose of excluding the node, thereby ensuring that the algorithm will not pass through the threat node.

D. Pheromone Update Strategy

Each time the iteration *n* is increased the pheromone on each path will volatilize once, the degree of volatilization of the pheromone expressed by the parameter $(1-\rho)$ and all ants complete an iterative cycle. The concentration of pheromones in each track segment is adjusted according to Eq. 7 and Eq. 8.

$$\left.\begin{aligned} \tau_{ji}(n+1) &= \rho\tau_{ji}(n) + (1-\rho)\Delta\tau_{ji} \quad \rho \in (0,1) \\ \Delta\tau_{ji} &= \Delta\tau_{ji}(n)^b \end{aligned}\right\} \tag{7}$$

Among them, $\Delta\tau_{ji}(n)^b$ represents the track corresponding to the minimum cost ant in the nth iteration different from the general algorithm, pheromone enhancement performed on the tracks of all ants. This paper only considers the pheromone enhancement of the minimum cost track in order to speed up the convergence of the algorithm, but in order to prevent the pheromone on some sides from growing too fast, the range of pheromone size is limited to an interval to avoid stagnation of the algorithm.

$$\Delta\tau_{ji}(n)^{bt} = \begin{cases} Q / L^b_N & j \in t_k(n); i \in t_k(n) \\ 0 & other \end{cases} \tag{8}$$

Among them, *Q* is a constant, which means that pheromone increases the intensity coefficient; L^b_N represents the minimum track cost in the nth iteration. The values of *Q* and ρ are determined according to the scale of the solution. The calculation stopped when the number of iterations is fixed or when the change of the solution is not obvious.

IV. PATH PLANNING SIMULATION AND RESULTS

In order to verify the performance of the proposed algorithm, this paper uses MATLAB for programming simulation. The threat cost from node *T* to target node *D* is high, so to track the optimize path and its parameters, see the Table 1.

Table I: Threat Area Point, Starting Point and Target Point Coordinates of Mobile Robot

	No.	Coordinates	No.	Coordinates
Threat point	1	(2,25)	7	(20,38)
	2	(11,29)	8	(21,27)
	3	(12,20)	9	(22,32)
	4	(13,30)	10	(24,19)
	5	(16,32)	11	(28,37)
	6	(16,39)	12	(37,34)

The simulation parameters of the algorithm are $n_{max} = 145$, $m = 12$, $\alpha = 1$, $\beta = 0.525$, $\gamma = 0.25$, $\rho = 0.15$, $Q = 50$, $R = 4$, $G = 2$, $L_m = 30$. The parameter n_{max} denotes the maximum number of iterations; L_{max} ensures that the battery consumption of the robot is within the allowable range. The Fig. 1 shows the optimal track, by using the proposed ant system strategy. It shows that the design algorithm shows better results as compared to ant system. The fitness calculation can be seen in Fig. 2 that the minimum cost track is constantly adjusted with the iteration and finally the global minimum cost track is obtained.

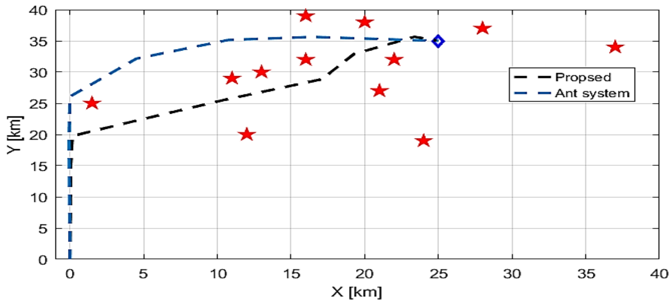

Fig. 1: Optimal Track under Different Weight Coefficients

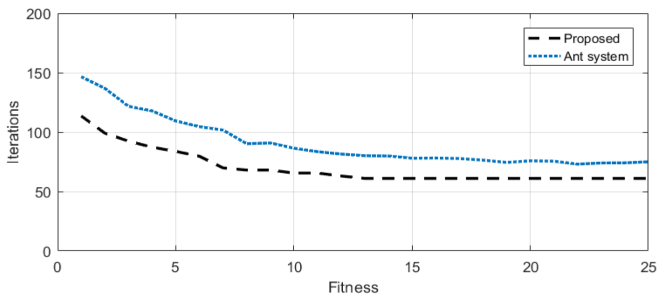

Fig. 2: Iteration of Optimal Individual Changing with Fitness

V. CONCLUSION

In this paper the ACO algorithm is applied for the path-planning problem of mobile robot. By solving major problems of path planning and introducing the guidance factor into the state transition strategy, the convergence speed of the algorithm guaranteed that the ant finally completes the track search. To set the current optimal path pheromone, update the strategy and setting the pheromone at the same time to improve the speed of the algorithm and at the same time prevents the algorithm from falling into a local optimum and stagnation. Simulation results show that the algorithm has reasonable results. Ants automatically find the target nodes in free space without setting navigation nodes and constructing VORONOI diagram and the convergence speed is fast. It overcomes the shortcomings of traditional ant system, i.e., the navigation nodes need to be set in advance and VORONOI diagram be constructed and has an encouraging application prospect in the field of track planning.

REFERENCES

[1] Hsu, C. H., & Juang, C. F. (2012). Evolutionary robot wall-following control using type-2 fuzzy controller with species-DE-activated continuous ACO. *IEEE Transactions on Fuzzy Systems, 21*(1), 100-112.

[2] Rashid, R., Perumal, N., Elamvazuthi, I., Tageldeen, M. K., Khan, M. A., & Parasuraman, S. (2016, September). Mobile robot path planning using Ant Colony Optimization. In *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)* (pp. 1-6). IEEE.

[3] Lanjewar, A., Sahare, V. N., & Sahare, N. (2010, February). An Approach based on clustering method for Object Finding Mobile Robots using ACO. In *2010 Second International Conference on Machine Learning and Computing* (pp. 161-165). IEEE.

[4] Martínez-Marroquín, R., Castillo, O., & Soria, J. (2009). Optimization of membership functions of a fuzzy logic controller for an autonomous wheeled mobile robot using ant colony optimization. In *Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control* (pp. 3-16). Springer, Berlin, Heidelberg.

[5] Yen, C. T., & Cheng, M. F. (2018). A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance. *Microsystem Technologies, 24*(1), 125-135.

[6] Liu, J. X., & ZHAO, Y. K. (2008). Application of ant colony algorithm and cellular automata model. *Journal of Nanjing University of Technology (Natural Science Edition)*, 6.

[7] Ganganath, N., & Cheng, C. T. (2013, October). A 2-dimensional ACO-based path planner for off-line robot path planning. In *2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (pp. 302-307). IEEE.

[8] Meng, Q. H., Yang, W. X., Wang, Y., Li, F., & Zeng, M. (2012). Adapting an ant colony metaphor for multi-robot chemical plume tracing. *Sensors, 12*(4), 4737-4763.

[9] Nearchou, A. C. (1998). Path planning of a mobile robot using genetic heuristics. *Robotica, 16*(5), 575-588.

[10] Marsland, S., Nehmzow, U., & Shapiro, J. (2000). Novelty detection on a mobile robot using habituation. From Animals to Animats: Proceedings of the *Sixth International Conference on Simulation of Adaptive Behaviour (SAB'00)*(pp. 189–198).