

# **Robot Global Path Planning Based on an Improved Ant Colony Algorithm**

# **Jingang Cao**

Department of Computer, North China Electric Power University, Baoding, China Email: caojg168@126.com

Received 9 January 2016; accepted 12 February 2016; published 15 February 2016

Copyright © 2016 by author and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY). http://creativecommons.org/licenses/by/4.0/ •

**Open Access** (cc)

# Abstract

Aiming at the disadvantages of the basic ant colony algorithm, this paper proposes an improved ant colony algorithm for robot global path planning. First, adjust the pheromone evaporation rate dynamically to enhance the global search ability and convergence speed, and then modify the heuristic function to improve the state transition probabilities in order to find the optimal solution as quickly as possible; and finally change the pheromone update strategy to avoid premature by strengthening pheromone on the optimal path and limiting pheromone level. Simulation results verify the effectiveness of the improved algorithm.

# **Keywords**

Path Planning, Ant Colony Algorithm, Grid Method

# 1. Introduction

The path planning is an important ability in many applications, such as UAV (Unmanned Aerial Vehicle), robotics, unmanned car and so on. Its task is to find a path from the current point (or the start point) to the target point, which is a shortest or a minimum price path without barrier in the environment which has obstacles.

The path planning is further divided into two categories [1]—global path planning and local path planning. Global path planning is required that environment should be completely known and the obstacles should be static. The robot produces a path from the starting point to the destination before it starts moving. But local path planning enables a robot to plan its path as it is moving in the environment, which means the robot will be able to generate new paths in response to the changes of environment [2].

In recent years, many researchers have studied the global path planning on various intelligent methods, such as genetic algorithms [3], particle swarm algorithms [4], neural networks [5] and ant colony algorithm [6], etc.

M. Dorigo, Italian scholar, who was inspired from ant foraging behavior and first proposed the ant colony al-

gorithm in 1991 [6]. Ant colony algorithm is a swarm intelligent algorithm, and it has many advantages: 1) strong solved robustness; 2) easy to implement parallel processing; 3) easy to combine with other algorithms to improve [7]. However, the basic ant colony algorithm also has many disadvantages: 1) easy to fall into local optimal solution; 2) large calculation, slow convergence and so on. The paper considers the problems of ant colony algorithm and uses it to the robot global path planning.

The paper is organized as follows: Section 2 and 3 describes the method of grid modeling and the basic ant colony algorithm; Section 4 describes the improved ant colony algorithm for the robot; Section 5 presents the experimental results on robot global path planning based on ant colony algorithms. Section 6 presents brief concluding remarks.

## 2. Grid Modeling

Grid method was proposed by the W. E. Howden in 1968, and its mainly task is to build a path grid map depending on the environment. The basic principle is to divide robot working environment into numerous tiny grid units, and the specifications of each grid is determined by the robot steps, namely one step is one grid unit. The grid is divided into two kinds: free gird and obstacle grid. Free grid is represented by white grid, and obstacle grid is represented by black grid. The grid map can be represented by a binary matrix, which 1 represents obstacle and 0 is free grid. The obstacle can occupy a grid or multiple grids, if it less than one grid is also expressed by one grid. Robot can only move in the free grid and must avoid when it encounter obstacles grid.

Depending on the position of grid, the grid can be divided into intermediate grid and boundary grid. For intermediate grid, robot may have eight directions for the next motion. Such as up, down, left, right, right-up, right-down, left-up and left-down. **Figure 1** shows the motion direction of the robot. And for boundary grid, it has to subtract inaccessible directions. Robot must avoid obstacles to select an optimization motion path moving to the target position.

The paper uses two-dimensional grid represents robot environment, and encodes the grid from left to right, from top to bottom. Figure 2 is a  $20 \times 20$  grid map.

In **Figure 2**, the number of grid in each row  $N_x = 20$ , and the number of grid in each column  $N_y = 20$ . The coordinate of the first grid is (0.5, 19.5), and the coordinate of the second grid is (1.5, 19.5), ... Suppose  $S = \{1, 2, 3, \dots, N\}$  is a set of the number of grids, the coordinate of the *i*th grid is  $(x_i, y_i)$  and  $R_a$  is the length of each unit in axis. The relation between the coordinate and the number of the grid can be expressed as follows:

$$\begin{cases} x_i = R_a \left\lfloor \mod(i, N_x) - \frac{R_a}{2} \right\rfloor \\ y_i = R_a \left[ N_y + \frac{R_a}{2} - ceil(i/N_y) \right] \end{cases}$$
(1)

where,  $N_x$  is the number of grid in each row;  $N_y$  is the number of grid in each column; *i* is the *i*th grid; *mod* is a function to get remainder, and *ceil* is a function to get integer.



Figure 1. The motion direction of the robot.



# 3. The Basic Ant Colony Algorithm

Ant colony algorithm (ACO) is an intelligent heuristic search algorithm by simulating the ant behavior to find the optimal path between the food source and their nest. Ants release pheromone on the path which they passed. Through these pheromones, ants can communicate with each other and find the shortest path to food finally. When the ant reaches an intersection at first time, it will randomly choose a motion direction forward. The more ants follow a given path, the more pheromone will be left on the path, and also the more attractive this path becomes to be followed by other ants. The probability of other ants selecting the path is increased. Ants can also adapt to environmental changes, when there is an obstacle on the path, they will find a new path quickly. This process can be expressed as a loop of positive feedback.

The probability of the ant *k* moving from the grid *i* to *j* is defined as follows [6]:

$$p_{ij}^{k}(t) = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{\substack{s \in \text{allowed}_{k}}} \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)} & j \in \text{allowed}_{k} \\ 0 & \text{otherwise} \end{cases}$$
(2)

where,  $p_{ij}^{k}(t)$  denotes the transition probability in which the ant k will traverse from grid i to j at time t;  $\tau_{ij}(t)$  is the intensity of the pheromone between grid i and j at time t;  $\eta_{ij}(t)$  represents the heuristic function between grid i and j;  $\alpha$  is information inspiration factor and  $\beta$  is hope inspiration factor; allowed<sub>k</sub> denotes a set of grids the ant k selecting in next step.

The process of robot path planning based on ant colony algorithm can be divided into two stages: optimization stage and stagnation stage. In optimization stage, the algorithm should have stronger ability of global search and can rapidly converge; and during stagnation stage, the algorithm should can automatically jump out of local optimal solution and continue to search the global optimal solution. To avoid premature to fall into local optimal solution and blind searching, the algorithm requires randomness of search and accuracy of solution. But randomness and accuracy are usually both contradictory and interrelation. In order to overcome defects of ACO, this paper presents an improved ant colony algorithm (IACO).

## 4. The Improved Ant Colony Algorithm

### 4.1. Optimize the Initial Pheromone

For the basic ant colony algorithm, the searching time is too long. One of the reasons is that the direction of the initial moment when the ants begin to search is uncertain. To improve the problem, and speed up the search, we

initialize pheromone decreasing with distance so that the ant has a clear motion direction during the initial search, the method is as follows:

$$\tau_{ig}(0) = \frac{C_1}{d_{ig}} \tag{3}$$

$$d_{ig} = \sqrt{\left(x_{i} - x_{g}\right)^{2} + \left(y_{i} - y_{g}\right)^{2}}$$
(4)

where,  $\tau_{ig}(0)$  denotes the initial pheromone level of a free grid  $i(x_i, y_i)$ ;  $C_1$  is a pheromone constant;  $d_{ig}$  represents the distance between the grid  $i(x_i, y_i)$  and the target position G  $(x_g, y_g)$ .

#### 4.2. Adjust Pheromone Evaporation Rate Dynamically

In ACO, pheromone evaporation rate  $\rho$  is unchanged and  $\rho \in (0,1)$ , which is directly related to the global search ability and convergence speed. If  $\rho$  is too small, ACO is prone to local convergence. On the contrary, if  $\rho$  is too large, although it can improved ACO random search performance and global search capability, the convergence rate of the algorithm will be reduced. So the paper adjusts  $\rho$  dynamically with the change of the number of cycles. In order to reduce the amount of calculation,  $\rho$  does not adjust in each cycle. In the early stage of algorithms,  $\rho$  has a large value to enhance the global search ability of the algorithm. With the number of cycles increases,  $\rho$  reduced appropriately, and the algorithm can converge to the optimal solution quickly. Specific rules are as follows:

$$\rho = 1 - \rho_0 \cdot \frac{\operatorname{ceil}(k/F)}{\operatorname{ceil}(k_{\max}/F)}$$
(5)

where, k represents the current number of cycle;  $k_{\text{max}}$  represents the maximum number of cycle;  $\rho_0$  is the initial value of the pheromone evaporation rate, F is the frequency factor, the bigger F is, the slower  $\rho$  will be adjusted.

#### 4.3. Improved Heuristic Function

In ACO, the heuristic function is:

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \tag{6}$$

where,  $d_{ij}$  represents the distance between grid *i* and the next grid *j*. But in grid map, the difference of value between adjacent grids is not obvious. In order to improve search efficiency of the algorithm, we use  $d_{jg}$  instead of  $d_{ij}$ , and  $d_{ig}$  represents the distance the next grid *j* to the target grid *g*.

In grids map, when the target position is known, a ant can calculate distances from its surrounding eight grids to the target point. The smaller the distance, the larger the value of  $\eta_{ij}$ . So the heuristic function can be expressed as

$$\eta_{ij}(t) = \frac{C_2}{d_{ig}} \tag{7}$$

where,  $C_2$  is heuristic factor, and it is a constant.

## 4.4. Modified Pheromone Update Strategy

Scholars have proposed several different pheromone update models, there are three mainly models: Ant-Cycle model, Ant-Quantity model and Ant-Density model. The paper chooses Ant-Cycle model, which can be expressed as follows:

$$\Delta \tau_{ij}^{k} = \begin{cases} \frac{Q}{L_{k}}, & \text{if ant} k \text{ use edge}(i, j) \text{ in its tour} \\ 0, & \text{otherwise} \end{cases}$$
(8)

where Q is a constant of pheromone,  $L_k$  is the length of path passed by the ant k, and  $\Delta \tau_{ii}^k(0) = 0$ .

All ants move one step called once iteration, after *n* iterations, all ants finish a cycle. In a cycle, there is a current optimal solution  $L_{best}$  and a worst solution  $L_{worst}$ . Using these two values to select some good solutions and update pheromone quantity, the rules are as follows:

$$L \le \frac{L_{best} + L_{worst}}{2} \tag{9}$$

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}$$
<sup>(10)</sup>

$$\Delta \tau_{ij}^{k} = \begin{cases} \frac{Q}{L}, & \text{if ant} k \text{ use edge}(i, j) \text{ in its tour} \\ 0, & \text{otherwise} \end{cases}$$
(11)

## 4.5. Limited Pheromone Quantity

When the pheromone quantity is too high, the algorithm is prone to premature, and it will reduce the optimization capability. The paper introduces Max-Min ant method to pheromone update strategy, which can be described as follows:

$$\tau_{ij} = \begin{cases} \tau_{\min}, \text{ if } \tau_{ij} \leq \tau_{\min} \\ \tau_{ij}, \text{ if } \tau_{\max} < \tau_{ij} < \tau_{\min} \\ \tau_{\max}, \text{ if } \tau_{ij} \geq \tau_{\max} \end{cases}$$
(12)

In Formula (12), the pheromone quantity is limited between  $[\tau_{\min}, \tau_{\max}]$ , which can make the difference of the pheromone quantity between the worst path and the optimal path is not too large.

#### 4.6. Algorithm Steps

Step 1: Initialization parameter: setting the start and target point, the maximum number of cycles  $k_{\text{max}}$ , the number of ant colony, and other parameters such as  $\alpha, \beta, \rho$ , etc.;

Step 2: Put all ants on the start point, and each ant determines which path it will select according to the transition probability equation;

Step 3: Save the path and the path length of each ant in each cycle;

Step 4: When one cycle finish, update pheromone quantity according to pheromone strategies;

Step 5: Loop execution Step 2 to 4 until the optimal solution is got or reach the maximum number of cycles.

#### 5. Simulation and Analysis

In order to verify the effectiveness of the improved ant colony algorithm, the paper simulates on Matlab 7.1 platform. The main parameters as follows:  $\rho = 0.7$ ;  $\alpha = 1$ ;  $\beta = 5$ ; Q = 1000.

To compare with the basic ant colony algorithm (ACO), we design the five different grid maps as **Figures 3-7**, which are the results of the simulation. **Figure 1** is a  $10 \times 10$  size and the simple and small map. **Figure 2** is a  $15 \times 15$  size and more complex than **Figure 1**. **Figures 3-5** is a  $20 \times 20$  size,  $25 \times 25$  size and  $30 \times 30$  size respectively. In different figures, there are two parameter were changed during simulation, they are the maximum number of cycles and the number of ants. In each Figure, (a) and (b) show the optimal path with IACO and ACO; (c) and (d) are the convergence curve about the average length of path and the number of cycle.

In the different five maps, each map simulates 20 times independently. **Table 1** shows the results of comparison between IACO and ACO. Form **Table 1**, we can see that the average number of cycles with IACO is much less than ACO, which nearly reduce one-third of the number of cycles. Also, the average length of the optimal path with IACO is shorter than ACO. It's because IACO improves the transition probability, and ants are prone to find the optimal path. In other words, the results show that global optimization ability and searching speed are improved obviously with IACO.



```
Figure 3. (a) IACO; (b) ACO; (c) IACO; (d) ACO.
```













(b)



Figure 5. (a) IACO; (b) ACO; (c) IACO; (d) ACO.













#### Figure 7. (a) IACO; (b) ACO; (c) IACO; (d) ACO

-		-	-				0				
	o b			'L'ho	100011	40	0 ±	the	0.0100	0.000	0.00
		Le.		1 110	T PST		())	THE.	(())))	ידואמ	senti
_	an		1.	1110	resu	LLD.	U1	unc	COIII	puii	5011

Tuna of mon	The average 1	number of cycles	The average length of the optimal path			
Type of map —	ACO	IACO	ACO	IACO		
Figure 3	31	21	15.7279	14.8995		
Figure 4	43	34	22.799	21.3848		
Figure 5	99	50	30.2668	29.1153		
Figure 6	153	95	37.0692	36.2885		
Figure 7	212	147	44.5269	43.1838		

# 6. Conclusion

This paper proposes an improved ant colony algorithm and applies it to the robot path planning. In the improved algorithm, pheromone quantity is reinforced in some short paths of each cycle, and pheromone evaporation rate is adjusted dynamically and so the transition probability is improved. The simulation results show that these measures are effective and can enhance global optimization ability, searching speed, and can avoid premature.

#### References

- Sedighi, K.H., Ashenayi, K., Manikas, T.W., Wainwright, R.L. and Tai, H.M. (2004) Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm. *Proceedings of the* 2004 *IEEE Congress on Evolutionary Computation*, 2, 1338-1345. <u>http://dx.doi.org/10.1109/CEC.2004.1331052</u>
- [2] Yee, Z.C. and Ponnambalam, S.G. (2009) Mobile Robot Path Planning Using Ant Colony Optimization. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM* 2009, Singapore, 14-17 July 2009, 851-856.
- [3] Zhou, L.F. and Hong, B.R. (2006) A Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot. *Chinese Journal of Acta Electronica Sinica*, **34**, 911-914.
- [4] Zhang, W.X., Zhang, X.L. and Li, Y. (2014) Path Planning for Intelligent Robots Based on Improved Particle Swarm Optimization Algorithm. *Chinese Journal of Computer Applications*, **34**, 510-513.
- [5] Deng, W.Y., Zheng Q.H., Chen, L. and Xu, X.B. (2010) Research on Extreme Learning of Neural Networks. *Chinese Journal of Computers*, 33, 279-287. <u>http://dx.doi.org/10.3724/SP.J.1016.2010.00279</u>
- [6] Dorigo, M., Maniezzo, V. and Colorni, A. (1996) Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Men and Cybernetics—Part B*, **26**, 29-41. <u>http://dx.doi.org/10.1109/3477.484436</u>
- [7] Wang, H.H. and Liu, Q. (2006) Convergence Analysis of a MAX-MIN ant Colony Algorithm. *Dynamics of Continuous Discrete and Impulsive Systems, Series B—Applications & Algorithms*, **13E**, 3901-3904.