# Path Planning for Indoor UAV Based on Ant Colony Optimization

Yufeng He[1], Qinghua Zeng[1], Jianye Liu[1], Guili Xu[1], Xiaoyi Deng[1]

1. The College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China
E-mail: heyufeng@nuaa.edu.cn

**Abstract:** UAV autonomous navigation is very useful in many applications, and path planning is one of the key technologies for UAV autonomous navigation. In this paper, the path planning problem to find the optimal path from the start location to the destination in an indoor environment is studied based on Ant Colony Optimization (ACO) algorithm. The workspace of UAV is modeled by applying the grid method, which is usually used in ground robot planning. With the help of 3D gird and new climbing weight parameter, a modified algorithm is presented to solve the premature convergence and low efficiency problem of traditional Ant Colony Optimization algorithm. The simulation results show that these improvements make the search of the optimal path rapidly and efficiently.

**Key Words:** UAV, Path Planning, Ant Colony Optimization

## 1 INTRODUCTION

UAV is a power-driven, unmanned, reusable aircraft which has advantages of good concealment, low cost and superior maneuverability. The number of applications using autonomous UAV is increased for civilian or military purposes. The urgent demand of some special applications such as target searching, information gathering, disaster relief, biochemical detection and emergency rescue promotes the rapid development of UAV which is able to work outdoors and indoors seamlessly. The technology to enable UAV for indoor navigation has attracted great attention.

Autonomous degree is an important index of UAV's performance. Indoor UAV with high-precision autonomous navigation function release controllers from complicated control work to focus on target search and identification. Path planning is one of the most important problems of autonomous navigation.

Path planning problem is to select an optimal collision-free path from a given start location to a destination in obstacle environment based on a certain evaluation criterion.

Few studies about UAV path planning is aimed at indoor environment. MIT's Robust Robotics Group had developed autonomous UAV capable of flying and avoiding obstacles in an indoor environment. They used Belief Roadmap (BRM) algorithm [1] to plan vehicle trajectories through the environment that incorporate a predictive model of sensing. In a lab at University of Pennsylvania, Vijay Kumar [2] and his team built flying quadrotors that could fly in an unknown indoor environment autonomously. They use topology method to carry out search-based path planning.

There are numerous researches on path planning algorithms for indoor ground robots. Many techniques are offered, such as genetic algorithm [3], particle swarm optimization [4], neural network [5], fuzzy logic [6], artificial potential field [7], visibility graph [8] and colony optimization [9][10]. Q.B. Zhu [11] applied ant colony algorithm to robot path planning by modeling the environment with grid method. The results demonstrate ant colony algorithm a good candidate for path planning problem. Could the mature methods developed for ground robots be applied to UAV path planning? This paper makes preliminary exploration on this issue.

## 2 ENVIRONMENT MODELING

The primary work of path planning is to establish the model of UAV's workspace. A good model of the environment information could improve the efficiency of path planning, and has good visibility in display.

The three-dimensional indoor workspace is broken down into several cells based on grid method. The sophistication of division affects the performance of the algorithm directly. More cells means more accurate of the workspace description, the planning result is more effective. But too many cells enlarge the workload greatly, hence decrease the entire efficiency.

According to laboratory scenes, this research divided the workspace into 10*10*5 cells. Each cell is a cube which has a side length as the step length of the UAV. Number the 500 grids. Figure 1 shows the number sequence of the bottom 100 grids. There are several static obstacles situated in the workspace. If there are any obstacles in a cell, the cell is viewed as an occupied one. The others are free cell. The schematic graph of the environment model is showed in Figure 2, in which the yellow parts are obstacles.

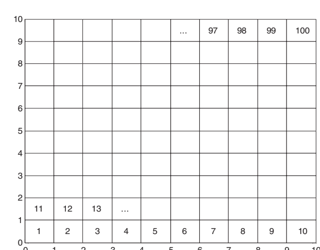


Fig. 1. The number sequence of the bottom 100 grids

This work is supported by National Nature Science Foundation under Grant No.61104188, 91016019, 61074162, 61174197, 61273057.

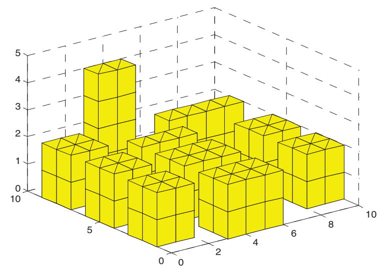


Fig. 2. The schematic graph of the environment model

To simplify the problem, we make assumptions as follows:

1） Store the information of all cells in three-dimensional matrix A. The corresponding matrix element of an occupied cell is 1, a free cell is 0. Store the numbers of occupied cells in matrix Ob.

2） The coordinates of each cell is represented by its center point. For example, the coordinate of the first cell is (1.5, 1.5, 1.5), the last cell is (9.5, 9.5, 4.5). The corresponding relations between the cell numbers and their coordinates are showed in formula (1) and (2).

$$\begin{cases} x = \mathrm{mod}(g-1,10)+0.5 \\ y = floor(\mathrm{mod}(g-0.1,100)/10)+0.5 \\ z = ceil(g/100)-0.5 \end{cases} \tag{1}$$

$$g=(z-0.5)\times 10\times 10+(y-0.5)\times 10+x+0.5 \tag{2}$$

3） The distance of two cells is calculated by the length of the line connecting the centers of them, denoted as $d(i,j)$, which is computed by formula (3).

$$d(i,j)=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2+(z_i-z_j)^2} \tag{3}$$

4） Wherever the UAV is in, the next cell to visit is adjacent to the current cell. It could be on any side: above, below, left, right, front, back, or diagonally adjacent. There are 26 directions to move at most.

## 3 METHODOLOGY

### 3.1 Basic Principle of ACO Algorithm

The ACO (Ant Colony Optimization) algorithm was first introduced by Dorigo in 1990s. It is an intelligent bionic algorithm imitates the foraging behavior of ants. Ants are social insects that live in colonies in which they cooperate with each other to find food sources. It was found that they communicate with each other by laying down a kind of aromatic essence called pheromone on the path they have travelled [12]. Assume there are two paths, path A and path B, are available for the ants to travel from their nest to the food sources. Path A is shorter than path B. Initially, as there is no pheromone trail on both paths, ants pick one path randomly. The ants on path A will reach the food resources earlier. When returning the nest, each ant reselects a path. The ants who select path A will always finish their travel in a shorter time. As a consequence, pheromone concentration on path A increases faster than that on path B. Other ants, observing the pheromone trail, are attracted to follow path A. Then pheromone concentrate on path A will be stronger therefore attracts more ants. Finally, as a result of the positive feedback, most ants select path A. Path A is selected to be the best path by the ant colony.

ACO algorithm shapes up by imitating the foraging process of ants described above. It has advantages of positive feedback of information, high parallelism, and optimization capability. Additionally, the foraging process of ant colony is quite similar with the path planning of UAV. Both of them are finding a best collision-free path from the start location to the destination. Thus the ACO algorithm is a good candidate to tackle the UAV path planning problem. Meanwhile, the algorithm has shortcomings such as low efficiency in the initial stage for the lack of pheromone, and easily trapped into local optimal solution.

### 3.2 Path Planning Based on ACO Algorithm

To simulate the foraging behavior of actual ant, consider the start location of UAV as the ant nest, and the destination of UAV as the food source. The process of UAV path planning based on ACO algorithm is to find the best path to the food source through the cooperation of ants.

Some notations and assumptions are presented as follows:

1） *Ant = {1， 2， …， k， …， m}* is the ant group, of which *m* is the amount of the ants and *k* is one of the ants.

2） Defined an array structure called tabu list for each ant to record the cells which have been visited. For each ant, revisit a cell in the list is forbidden. Another function of the list is to calculate the length of path when an ant finishes its travel. The tabu list is notated as *ant (i).visited (i=1， 2， …， m).*

3） Array *allowed* is the candidates group for the next step which varies as the location changes. Every time when an ant reach a new cell, search the adjacent cells, those cells, except the occupied ones and cells in tabu list, are the candidates.

4） Pheromone level of any edge is stored in matrix $\tau$ . $\tau(i,j)$ indicates the pheromone level on edge between cell i and cell j and could be notated as $\tau_{ij}$. All elements are initialized at $\tau_0$ .

Put m ants on the start cell. Each ant takes the current cell as a center and then selects a cell for the next step from their selectable cells group based on a given criterion. The criterion is called random proportional rule. At time t, the probability for ant k selecting the path from cell i to j is calculated by formula (4).

$$p_{ij}^k(t)=\begin{cases} \dfrac{\tau_{ij}{}^{\alpha}(t).\eta_{jg}{}^{\beta}(t)}{\sum_{s\in allowed}\tau_{is}{}^{\alpha}(t).\eta_{sg}{}^{\beta}(t)}, \mathrm{if}\ \ j\in allowed \\ 0 \qquad\qquad ,\mathrm{otherwise} \end{cases} \tag{4}$$

Where $\tau_{ij}(t)$ is the pheromone level on edge (i, j), $\eta_{jg}(t)$ is the visibility from cell j to the terminal cell and defined as $\eta_{jg}(t)=1/d(j,dest)$ . Parameter $\alpha$ and $\beta$ determine the relative importance of pheromone level and visibility respectively.

In ACO algorithm, another criterion is used as the path selecting strategy which is called pseudo-random proportional rule. This rule is showed in formula (5).

$$j = \begin{cases} \arg \max_{j \in allowed} \{ \tau_{ij}^{\alpha}(t).\eta_{ij}^{\beta}(t) \}, & if\ q \le q_0 \\ S & , otherwise \end{cases} \tag{5}$$

Where $q_0 \in (0,1)$ is a pre-set constant, $q \in (0,1)$ is a random number. If $q \le q_0$, choose cell j which maximize $p_{ij}^k(t)$, otherwise choose a cell based on Roulette Algorithm. After an ant traversing an edge (i, j), update the pheromone level on this edge as formula (6).

$$\tau_{ij}(t+1) = (1-\xi)\tau_{ij}(t) + \xi \cdot \Delta\tau_{ij}^k \tag{6}$$

Where $\xi$ is the local evaporation coefficient of pheromone, and $\Delta\tau_{ij}^k$ is a constant which could be set at the initial value of pheromone level.

An iteration cycle is completed when all ants have reached the destination. Update pheromone level on all edges according to formula (7) and formula (8).

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t,t+1) \tag{7}$$

$$\Delta\tau_{ij}(t,t+1) = \sum_{k=1}^{m} \Delta\tau_{ij}^k(t,t+1) \tag{8}$$

Where $\rho \in (0,1)$ is the global evaporation coefficient of the pheromone. Evaporation coefficient prevents the solution from trapping into a suboptimal solution, thus guarantee the chance to search for new solution. $\Delta\tau_{ij}(t,t+1)$ is the increment of pheromone on edge *(i, j)* while $\Delta\tau_{ij}^k(t,t+1)$ is part of it laid down by ant k.

The expression of $\Delta\tau_{ij}$, $\Delta\tau_{ij}^k$ differs in different specific method. There are three models given by Dorigo: ant-cycle sytem, ant-quantity and ant-density system. Numerous studies show that the ant-cycle model is superior to the other two models. In any-cycle model, the pheromone laid down by ant k is calculated by formula (9).

$$\Delta\tau_{ij} = \begin{cases} Q / L(k), & if\ (i,j)\ belongs\ to\ the\ path\ of\ ant\ k \\ 0 & , otherwise \end{cases} \tag{9}$$

Where *L(k)* if the length of the path ant k has visited in this iteration.

Put all ants on the start cell and start next iteration after updating the pheromone table and recording the result of this iteration. The exploration is repeated until stop criteria is met. The stop criteria could be a preset iteration times or a condition concerning the performance of the solution.

### 3.3 Modified algorithm

The ACO algorithm has significant advantages. But its shortcomings cannot be ignored. For instance, the solving speed is very low at initial stage for the lack of guidance from pheromone, and it is easily trapped into local optimal solution. To solve these problems and take into account the energy consumption of UAV, a modified algorithm is presented.

In ACO algorithm, each ant constructs its solution in a greedy search algorithm. Every ant is able to find a solution, maybe a poor solution. But the ant colony could find an optimal solution through the collaboration of all individuals in the group. In order to keep ants searching separately in parallel in an iteration cycle, local update of pheromone is not executed in the algorithm presented in this paper.

The mechanism of positive feedback ensures the solution of the problem evolving in a direction toward the global optimal solution. However, in some cases, the positive feedback causes premature convergence phenomenon. For instance, due to the existence of a local optimal solution, or simply because of the initial random oscillation, a suboptimal individual makes a strong influence upon the group, preventing the group for further exploration in the solving space.

In this algorithm, the premature convergence manifests as convergence to a local optimal solution. Basic reason of this phenomenon is that the probability distribution showed in formula (4) is determined by pheromone level and visibility. The visibility is defined as the reciprocal of the distance between candidate cell and destination cell. So ants tend to select a cell closer to the destination, therefore may miss a path which is not directly heading the destination in a period but is shorter in total length. Meanwhile, a path which always heads to cells that have highest probability, this is what we call the local optimal solution, will be picked by majority of ants. As the difference of length of the two paths is not quite large, the superiority of the shorter path could not be reflected obviously. The pheromone on the local optimal path accumulates quickly. Algorithms eventually converge to this undesired path because of constant positive feedback.

To solve the problem, the pheromone updating rule is modified. In each iteration cycle, the pheromone is only updated on the best path. So even if there is a local optimal path, it will not make great difference as long as one of the m ants could find a shorter path. Simulation results showed that this strategy prevents the algorithm from being trapped in local optimal solution effectively and convergence speed is improved.

Based on the former algorithm, the search behavior of ants will quickly concentrate around the optimal path, thus improve the efficiency of the algorithm. But it brings a new problem: with the strengthening of positive feedback, pheromone level on the edges which belong to the optimal solution is much higher the rest. The following ants will repeat to construct the same solution instead of searching for new solution. This causes a waste of time and brings a possibility of trapped in another suboptimal solution.

To avoid such situation, the pheromone levels are limited between $\tau_{min}$ and $\tau_{max}$, as in the MAX-MIN Ant system described in [13]. The pheromone levels for edges are initialized at $\tau_{max}$, so that the difference between good solution and bad solution solution will not increase too fast thus ants could explore for more different solution.

In addition, taking into account the energy consumption of UAV, a climbing weight is introduced to the length of path. The addition of weighting makes ants select the best direction to bypass obstacles, horizontally or from the top.

Variable h is introduced to record the climbing weight of each ant. It's combined with path lengh L(k) to form a cost function as formula (10).

$$W(i) = L(i) + h \times k$$

Where $k$ is a coefficient determines the importance of $h$.

The flowchart of the algorithm presented in this paper is given in figure 3.

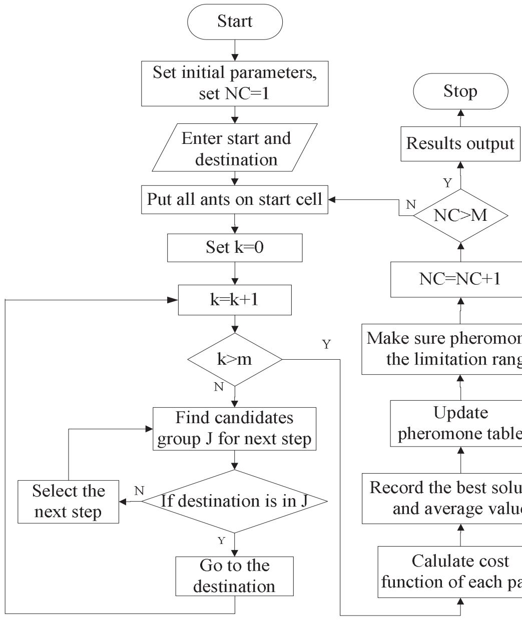


Fig 3. Flowchart of the modified algorithm

## 4 SIMULATION AND RESULTS

In this section, simulations of the proposed algorithm are performed in MATLAB. The workspace is modeled in a grid map with 10*10*5 grids as shown in figure 2. Parameters used in simulations are given in Table 1. Input the number of start and terminal cell at the beginning of simulation. The result of UAV path planning is presented by three figures: the optimal path, the cost function of iteration-optimal path, the average cost function in each iteration cycle. The optimal path figure shows the path in the environment directly while the other two figures show the performance of algorithm.

Table1. Parameters Used in Simulations

| Parameters | Value |
|---|---|
| Ants Amount ($m$) | 31 |
| Iterations (M) | 50 |
| Coefficient ( $\alpha$ ) | 1 |
| Coefficient ( $\beta$ ) | 5 |
| Evaporating Rate ( $\rho$ ) | 0.3 |
| Coefficient ( $Q$ ) | 100 |
| Coefficient ( $q_0$ ) | 0.5 |
| Weighting ( $k$ ) | 1 |

Two results are given here in figure 4~figure 10. The start and terminal cells in the two cases are (18, 163) and (7, 92) respectively.

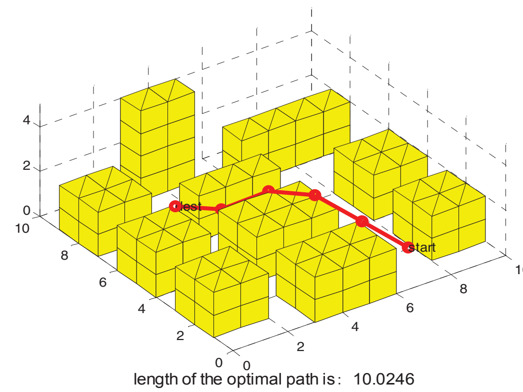


Fig 4. The optimal path in case 1

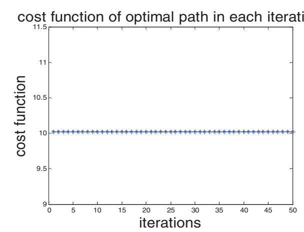


Fig 5. Cost function of iteration-optimal path in case 1

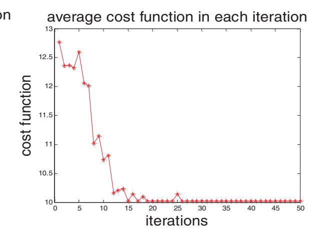


Fig 6. Average cost function in Case 1

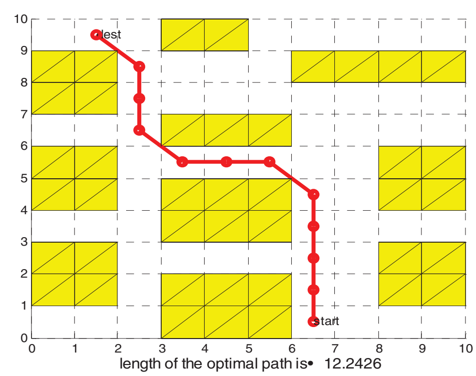


Fig 7. The optimal path in case 2. The top view is given because the whole path is at same height. This result shows that the UAV chooses to bypass the obstacles horizontally under the action of climbing weight.

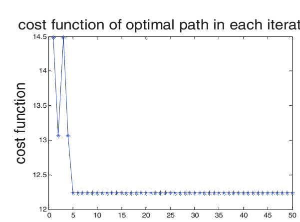


Fig 8. Cost function of iteration-optimal path in case 2

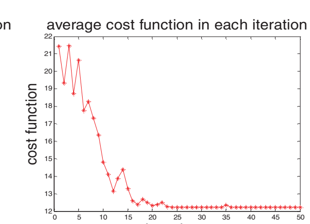


Fig 9. Average cost function in case 2

## 5 CONCLUSION

This paper presents an approach to path planning for indoor

UAV based on Ant Colony Optimization. To tackle the problem of premature convergence and improve the efficiency of exploration, some modifications to Ant Colony System are suggested. The energy consumption of UAV is taken into account by introducing a weighting of climbing height. Simulation results show that algorithm presented in this paper could produce good solution in much less time. Future work intends to smooth the path and deal with the boundary problem considering the size of UAV.

## REFERENCES


[1] Ruijie He, Sam Prentice, Nicholas Roy, Planning in Information Space for a Quadrotor Helicopter in a GPS-denied Environment, Proceedings of the IEEE International Conference on Robotics and Automation, 2008, 1814-1820.

[2] Subhrajit Bhattacharya, Vijay Kumar, Maxim Likhachev . Search-based Path Planning with Homotopy Class Constraints. In Proceedings of The Third Annual Symposium on Combinatorial Search. Atlanta, Georgia, 2010, 8-10.

[3] I.K. Nikolos, K.P. Valavanis, N.C. Sourveloudis, A.N. Kostaras, Evolutionary algorithm based offline/online path planner for UAV navigation, IEEE Transactions on System, Man, and Cybernetics – Part B. Cybernetics 33(6), 2003, 898-912,.

[4] M. Saska, M. Macas, L. Preucil, L. Lhotska, Robot path planning using particle swarm optimization of Ferguson splines, in: Conf. Emerging Technologies and Factory Automation, IEEE, 2006, 833 – 839.

[5] K. Althoefer, Neuro-fuzzy motion planning for robotic manipulators, Ph.D. thesis, King’s College, London, 1997.

[6] T.P. Fries, Fuzzy genetic motion planning under diverse terrain conditions for autonomous robots, in: Proc. Circuits, Signals, and Systems, 2004, 504 – 508.

[7] Huang Bingqiang, Cao Guangyi, The Path Planning Research for Mobile Robot Based on the Artificial Potential Field. Computer Engneering And Applications, Vol.42, No.27, 2006, 26-28.

[8] Yang Bing, Liu Weijie, A Improved Method of Robot's Path Planning Based Visibilty Graph. Computer Knowledge and Technology, Vol.5, No.2, 2009, 434-435.

[9] X. Fan, X. Luo, S. Yi, S. Yang, H. Zhang, Optimal path planning for mobile robots based on intensified ant colony optimization algorithm, in: Proc. of IEEE Int. Conf. Robotics, Intelligent Systems and Signal Processing, vol. 1, 2003, 131 – 136.

[10] Yinling Zhang, Xiaomei Niu. Simulation Research on Mobile Robot Path Planning Based on Ant Colony Optimization. Computer Simulation, Vol.28, No.6, 2011, 231-234.

[11] Qingbao Zhu, Yulan Zhang, An Ant Colony Algorithm Based on Grid Method for Mobile Robot Path Planning, Robot, 2005, Vol.27, No.2, 2005, 132-136.

[12] Rutger Claes, Tom Holvoet. Ant Colony Optimization applied to Route Planning using Link Travel Time predictions. IEEE International Parallel & Distributed Processing, 2011, 358-365.

[13] T. Stutzle, H. Hoos et al., Max-min ant system, Future Generation Computer Systems, Vol.16, No.8, 2000, 889–914.