International Journal of Computer Science & Applications © 2007 Technomathematics Research Foundation
Vol. 4 Issue 3, pp 57-68

Improved Ant Colony Algorithm for Global Optimal Trajectory Planning of UAV under Complex Environment

Guanjun Ma **Haibin Duan** **Senqi Liu**
gjma@asee.buaa.edu.cn **hbduan@buaa.edu.cn** **forest833@gmail.com**

School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, P. R. China

Abstract

A novel type of Ant Colony Algorithm (ACA) for the globally optimal trajectory planning of Unmanned Aerial Vehicle (UAV) is proposed in this paper. The parallelism and positive feedback of ACA is feasible in UAV trajectory planning under complex environments, but the basic ACA model has the limitation of stagnation, and easy to fall into local optimum. Hybrid improvement strategies for the basic ACA model are proposed in this paper, and a type of trajectory smoothing scheme is also put forward. Simulation results show that the improved ACA is effective and can be used in the real-time trajectory planning of UAV. It has also been verified that the proposed method has better performance in convergence speed, solution variation, dynamic convergence behavior, and computational efficiency than the UAV trajectory planning method based on the basic ACA model under complex environments.

Keywords: Ant Colony Algorithm (ACA), Unmanned Aerial Vehicle (UAV), trajectory planning, pheromone, positive feedback

1 Introduction

Unmanned Aerial Vehicle (UAV) is an aircraft with no onboard pilot. UAV can be remote controlled or fly autonomously based on pre-programmed flight plans or more complex dynamic automation systems. In the 21st century, military conflicts ask for an “intelligent” concept of operations with regard to intelligence gathering. A highly technological UAV platform, which can be deployed in expeditionary forces at all levels of conflict, satisfies this necessity. The requirements to boost the UAV technology and specifically the platform survivability, demands for flexibility and versatility in UAV deployment increase more and more. In advance of the deployment, an efficient and systematic planning process, which can cope with a complex environment, provides a substantial contribution to the improvements in effectiveness and mission outcome [1]. Automated trajectory planning is an integral and critical component of advanced mission planning systems. The goal of the trajectory planner is to compute within an appropriate time window, an optimal or near-optimal path for surviving penetration through a hostile enemy environment while satisfying mission objectives. The planner considers terrain data, threat information, fuel constraints, time constraints, and other constraints specified by the pilot or other aircraft subsystems. Typically, a trajectory planner will return waypoint locations and ETAs (estimated time of arrivals), headings, resource utilization (e.g., weapons, fuel consumption, sensor constraints), and metrics indicating the characteristics of the route such as risk and effectiveness. This path is then passed to a lower-level tactical route planner that uses models of the ownership and threat capability to generate a detailed flyable route. Trajectory planning is the challenging technology of UAV. Trajectory planning is the guidance of a specific agent from a source location to a destination whilst avoiding all encountered obstacles [2]. Among various trajectory planning approaches, some conventional techniques (e.g., genetic algorithm [3], or neural networks [4]) assume that all the threats in battle field have the same threat grade. However, different threats may impose different threatening effects on UAV under real air-battle environment, thus, new development on trajectory planning algorithm which enables UAV

automatically handle each potential threat is essential to meet the practical battlefield requirements.
The flight trajectory planning in a large mission area is a typical large scale optimization problem, This paper uses the Ant Colony Algorithm (ACA) to optimize the flight trajectory. ACA has been inspired by the observation on the real ant colony's foraging behavior and the shortest path between food source and their nest can be found on those ants. The principle of this phenomenon is that ants deposit a chemical substance (called pheromone) on the ground, thus, they mark a path by the pheromone trail. In this process, a kind of positive feedback mechanism is adopted. An ant encountering a previously laid trail can detect the dense of pheromone trail. It decides with high probability to follow a shortest path, and reinforces the trail with its own pheromone. The larger amount of the pheromone is on a particular path, the larger probability is that an ant selects that path and the path's pheromone trial will become denser. At last, the ant colony collectively marks the shortest path, which has the largest amount of pheromone. Such simple indirect communication way among ants embodies actually a kind of collective learning mechanism. ACA has successfully used in parameter optimization [5], continuous space optimization [6], fault diagnosis [7] and so on. In this paper, the basic ant colony algorithm is improved properly, and measures of keeping optimization, adaptively selecting and adaptively adjusting are applied in order to make the ant colony algorithm adapted well

The remainder of this paper is organized as follows. Section 1 introduces the importance of trajectory planning in the UAV. Subsequently, the fundamentals of the basic ACA algorithm are presented in Section 2. Then, in Section 3, we propose the cost function. Then, Section 4 describes the improved ACA algorithm approach to UAV trajectory planning. Experimental results are given in Section 5 to verify the feasibility and effectiveness of the proposed improved method. Our concluding remarks are contained in Section 6. Section 7 is acknowledgement.

2 Fundamentals of Ant Colony Algorithm

Ant Colony Algorithm (ACA) algorithm is a meta-heuristic algorithm for the approximate solution of combinatorial optimization problems that has been inspired by the foraging behavior of real ant colonies [8~10]. In ACA algorithm, the computational resources are allocated to a set of relatively simple agents that exploit a form of indirect communication mediated by the environment to construct solutions to the finding the shortest trajectory from ant nest to a considered problem. Real ants are capable of food source, because, while walking, ants deposit pheromone on the ground, and real ants have a probabilistic preference for trajectory with larger amount of pheromone. The natural metaphor on which ant algorithms are based is that of ant colonies. Real ants are capable of finding the shortest trajectory from a food source to their nest, without using visual cues by exploiting pheromone information. While walking, ants deposit pheromone on the ground, and follow, in probability, pheromone previously deposited by other ants. A way ants exploit pheromone to find a shortest trajectory between two points is shown in Figure. 1.
Consider Figure. 1(a): Ants arrive at a decision point in which they have to decide whether to turn left or right. Since they have no clue about which is the best choice, they choose randomly. It can be expected that, on average, half of the ants decide to turn left and the other half to turn right. This happens both to ants moving from left to right (those whose name begins with an *L*) and to those moving from right to left (name begins with a R). Figure. 1(b) and Figure. 1(c) show what happens in the immediately following instants, supposing all ants walk at approximately the same speed. The number of dashed lines is roughly proportional to the amount of pheromone that the ants have deposited on the ground. Since the lower trajectory is shorter than the upper one, more ants will visit it on average, and therefore pheromone accumulates faster. After a short transitory period the difference in the amount of pheromone on the two trajectory is sufficiently large so as to influence the decision of new ants coming into the system (this is shown by Figure. 1(d)). From now on, new ants will prefer in probability to choose the lower trajectory, since at the decision point they perceive a greater amount of pheromone on the lower trajectory. This in turn increases, with a positive feedback effect, the number of ants choosing the lower, and shorter, trajectory. Very soon all ants will be using the shorter trajectory. This process is thus characterized by a positive feedback loop, where the probability with which an ant chooses a trajectory increases with the number of ants that previously chose the same trajectory. With the above positive feedback mechanism, all ants will choose the shorter trajectory in the end.

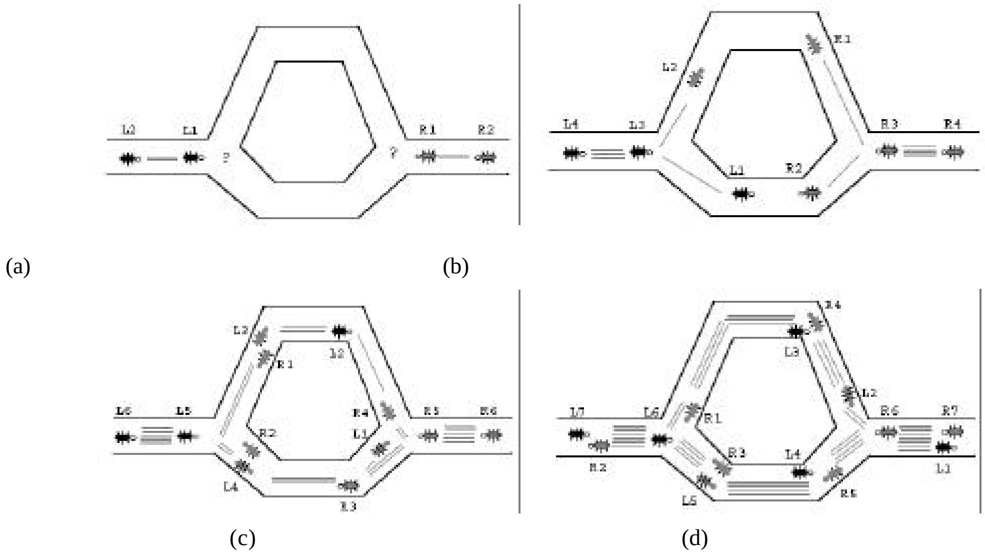

Figure 1: How real ants find a shortest trajectory

The above behavior of real ants has inspired ACA algorithm, an algorithm in which a set of artificial ants cooperate to the solution of a problem by exchanging information via pheromone deposited on graph edges. ACA algorithm has been applied to combinatorial optimization problems such as the traveling salesman problem (TSP), quadratic assignment problem (QAP), and so on.

3 Problem Statements

3.1 Assumptions

The vehicle is provided with basic autopilot functions, it maintains altitude by means of an auto-throttle, and airspeed is maintained by means of a pitch autopilot. Prevalent wind information is available [11]. Therefore, the analytical assumptions are as follows:
(1) The vehicle maintains altitude.
(2) The airspeed is constant.
(3) Average wind speed and direction are known.

3.2 The cost function of the flight trajectory

In order to illuminate the method, without loss of generality, the hostile threats are simply taken as enemy monostatic radars. The radar equation is given by [12]:

$$Q_i = \frac{P_t G A_e s}{(4p)^2} \qquad (1)$$

Where P is the power received by radar, P_t is power of the transmitter, G is gain of the antenna, A_e is effective acreage of the antenna, s is section acre age of radar, d is the distance from radar to object.

A network is constructed from the known threat locations. The intersecting point is a feasible waypoint for UAV. Each edge of the network is assigned two costs: threat cost and fuel cost. Threat costs are based on a UAV's exposure to all the enemy radars. An exact threat cost calculation would involve the integration of the cost along each edge. According to the radar equation (1), the ratio of the received Radio Frequency (RF) power to the transmitted RF power reflected from the target is inversely proportional to d^4, where d is the slant range from the UAV to the monostatic radar. So, the cost to be minimized is:

$$W = \sum_{i=1}^{N} \int_0^t \frac{Q_i}{d_i^{\,4}(t)} dt \qquad (2)$$

$Q_i = \frac{P_t G A_e s}{(4p)^2}$ is the detecting power of the i-th, N is the number of the radars, the missiles and other threats.

A computationally more efficient and acceptably accurate approximation to the exact solution is to calculate the threat cost at several locations along an edge and take the length of the edge into account. In this work, the threat cost was calculated at five points along each edge, as shown in Figure 2:

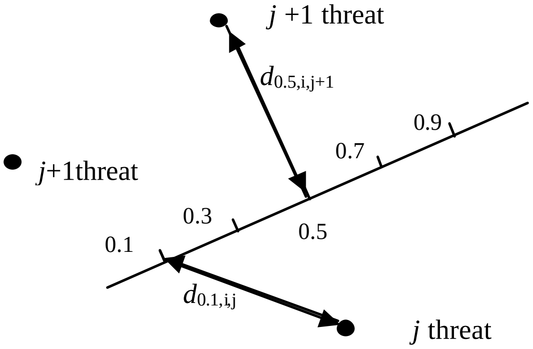

Figure 2: Computation of threat cost

The threat cost associated with the i-th edge is given by the expression [13]

$$w_{t,i} = L_i \sum_{j=1}^{N} \left(\frac{1}{d^4_{0.1,i,j}} + \frac{1}{d^4_{0.3,i,j}} + \frac{1}{d^4_{0.5,i,j}} + \frac{1}{d^4_{0.7,i,j}} + \frac{1}{d^4_{0.9,i,j}}\right) \tag{3}$$

Where $d^4_{0.1,i,j}$ is the distance from the 1/10 point on the i-th edge to the j-th threat, L_i is the i-th side length, N is the number of the radars, the missiles and other threats. Moreover, it can simply consider the fuel cost w_f equals to L, then $w_{f,i}$ equals to L_i when the speed is a constant. The total cost for traveling along i-th edge comes from a weighted sum of the threat and fuel costs:

$$w_i = k \cdot w_{t,i} + (1-k) w_{f,i} \qquad (0 \le k \le 1) \tag{4}$$

The choice of k between 0 to 1 gives the designer certain flexibility to dispose relations between the threat exposition degree and the fuel consumption. When k is more approaching 1, it asks to plan a shorter trajectory and pays less attention to the radar's exposed threat. Otherwise, when k is more approaching 0, it requires to avoid the threat as far as possible on the cost of sacrifice the trajectory length. In this paper, coefficient k is simply taken as 0.8.

3.1 Trajectory smoothing

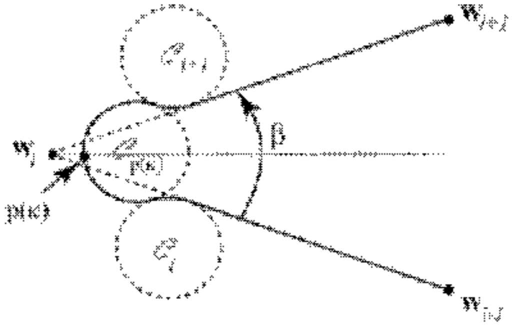

Figure 3: The k-trajectory smoothing result

Considering the UAV's turn constraint, the searched trajectory may not satisfy the demand, so we must smooth the trajectory where two nodes of network connect with each other. A k-trajectories smoothing method is adopted in this paper, as shown in Figure 3. Let $p(k)$ denotes a parameterized point on the line between w_i and p, on the bisector of the angle.

$$p(k) = w_i + kR\left(\frac{1}{\sin(\frac{b}{2})} - 1\right)\bar{q} \tag{5}$$

Where $k \in [0,1]$ .Clearly $p(0)=w_i$, $p(1)=p$, and $\bar{q} = \frac{q_{i+1} - q_i}{\|q_{i+1} - q_i\|}$. The detailed computation can be found in reference [14].

4 Path planning based on improved ACA

The ACA mainly contains two basic steps: adaption stage and cooperation stage. In the adaption stage, the candidate solutions continues to readjust its structure on the basis of information accumulating , In the cooperation stage, the candidate solutions exchange information to produce better solutions [9]. In ACA, an ant left pheromone which can be felt by the next ant as a signal to affect its action. The pheromone that the following one left will enhance the original pheromone and so circulation gets down. Thus, the more ants a trajectory is passed by, the more possible a trajectory is selected by the other ants. In certain time, a shorter trajectory will be visited by more ants, thus it will accumulate more pheromone and the possibility which is selected by other ants is bigger next time. This process can guarantee nearly all ants walk along the shortest trajectory.

4.1 The rule of choosing node

Suppose $\Gamma = \{t_{ij}(t) \mid c_i, c_j \subset C\}$ be the set of pheromone trail intensity on edge t_{ij} at time t. At the initial stage, we set $t_{ij}(0) = const$. We define the transition probability from node *i* to node *j* for the *k*-th ant as follows [15]:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[t_{ij}(t)]^a \cdot [h_{ij}(t)]^b}{\sum\limits_{s \subset allowed_k} [t_{is}(t)]^a \cdot [h_{is}(t)]^b} & \text{i f} \quad j \in allowed_k \\ 0£ \quad ¬ & else \end{cases} \tag{6}$$

Where *allowed*$_k$={*N*-tabu$_k$}, *a* and *ß* are parameters that control the relative importance of trail versus visibility, *?*$_{ij}$ is the heuristic desirability, and $h_{ij}(t) = 1/d_{ij}$. where d_{ij} is the distance between node *i* and node *j*,t_{ij} is the amount of pheromone trail on edge (*i*, *j*).

4.2 The rule of adjusting pheromone

When the ants move at the network, the pheromone level on the selected edge (*i*, *j*) is updated according to the local updating rules in Eq. (7) and Eq. (8). In this way, ants will make better use of their pheromone trail, without local updating, all ants will search in a narrow neighborhood of the best previous tour.

$$t_{ij}(t+n) = (1-V) \cdot t_{ij}(t) + V \cdot t_0 \tag{7}$$

$$t_0 = \frac{1}{n l_{nn}} \tag{8}$$

Where V is the local pheromone decay parameter, and l_{nn} denotes the nearest distance between two nodes in the network. Once all ants completed the iteration (to find a feasible trajectory), global pheromone updating takes place. This can be expressed in Eq. (9) and Eq. (10).

$$t_{ij}(t+n) = (1-r) \cdot t_{ij}(t) + \Delta t_{ij}(t) \tag{9}$$

$$\Delta t_{ij}^k(t) = \begin{cases} \dfrac{Q}{W_k} \text{,if the kth ant uses edge (i, j) in its tour} \\ \text{0, otherwise} \end{cases} \tag{10}$$

Where r is the global pheromone decay parameter to weaken the original information, and $r \subset [0,1]$. Where *Q* is a constant and affects the algorithm's convergence rate in certain degree, *W*$_k$ is the generalized cost of the *k*-th ant that it tours in this cycle.

4.3 The improvement of the algorithm

The ACA algorithm usually converges at the local least value and the convergence speed is slow. In order to overcome these shortcomings, improve the whole searching capability, several strategies are proposed in this sub-section.

(1) The ACA algorithm uses the elitist strategy: reserve the optimum result of algorithm. At the end of each iteration, reserve its optimum solution,

(2) Independent of the choice between the iteration-best and the global-best ant for the pheromone trail update, search stagnation may occur. Such a stagnation situation should be avoided, One way for achieving this is to influence the probabilities for choosing the next solution component, which depends directly on the pheromone trails and the heuristic information. The heuristic information is typically problem-dependent and static throughout the application of the algorithm. But by limiting the influence of the pheromone trails one can easily avoid the relative differences between the pheromone trails during the employment of the algorithm. To achieve this goal, ACA imposes and explicitly limits $t_{\min}$ and $t_{\max}$ on the minimum and maximum pheromone trails such that for all pheromone trails, so at any time $t_{ij}(t) \subset [t_{\min}, t_{\max}]$. This can be expressed as follows [16]:

$$t_{ij}(t+n) = \begin{cases} t_{\min}, & if \quad t_{ij}(t) < t_{\min} \\ t_{ij}(t), & if \quad t_{\min} < t_{ij}(t) < t_{\max} \\ t_{\max}, & if \quad t_{ij}(t) > t_{\min} \end{cases} \tag{11}$$

(3) When the problem scale is relatively large, if the global pheromone decay parameter *?* is too big, the global convergence ability of ACA will degrade. If the global pheromone decay parameter *?* is too small, the global convergence ability will be enhanced, while the convergence speed of ACA will become slow. Therefore, self-adaptive control strategy is adopted for the variation of the global pheromone decay parameter *?*, the initial value of could be a big value, such as *?*(0)=0.9.With the increase of iteration number, if the optimum value varies little, which show the iteration process that fall into the local best, rather than the global best. On this occasion, the value *?* should change according to the following function [17]:

$$r = \begin{cases} 0.95 \cdot r(t-1) & r \geq r_{\min} \\ r_{\min} & r < r_{\min} \end{cases} \tag{12}$$

4.4 Programming Steps of the Improved ACA

The programming steps of the improved ACA in solving trajectory planning can be described as follows:

Step 1 Initialize the pheromones of all nodes and place all ants at the start point.

Step 2 Every ant chooses the node according to Eq.(6) , and reach the goal point, then form a feasible trajectory at last ,

Step 3 *Nc* = *Nc*+1, calculate cost function value W of each ant and reserve the current optimal solution, update the pheromone on each way according to formula (7) ~ (11).

Step 4 Examine the optimized solution, and decide if pheromone evaporation gene ? needs to be adjusted. If it needs, the pheromone evaporation gene ? will be adjusted according to formula (12).

Step 5 Judge if it satisfies the iterative condition Nc > Ncmax, .If it satisfies, end the iteration and output the best solution. Otherwise, return to **Step 2** until satisfy the iterative condition.

The flowchart in Figure.4 describes the above-mentioned procedure.

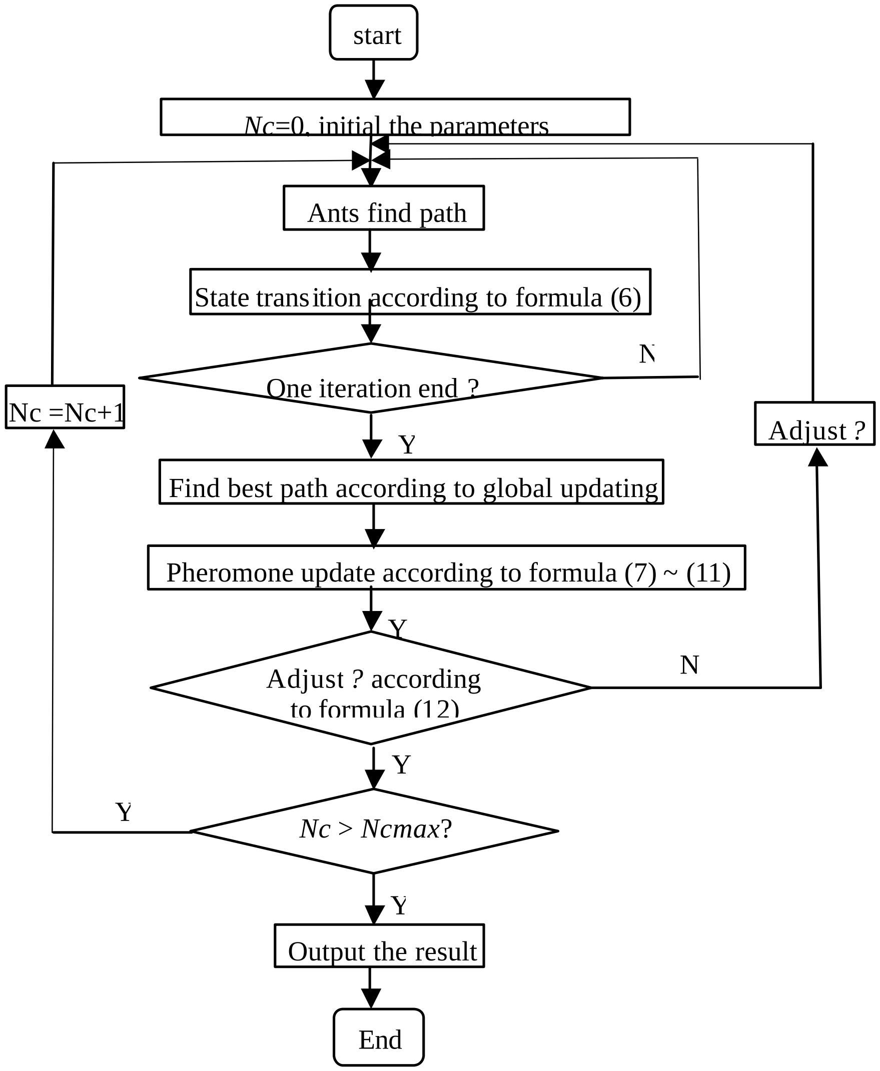
start
Nc=0, initial the parameters
Ants find path
State transition according to formula (6)
One iteration end ?
N
Y
Nc =Nc+1
Adjust ?
Find best path according to global updating
Pheromone update according to formula (7) ~ (11)
Y
Adjust ? according to formula (12)
N
Y
Y
Nc > Ncmax?
Y
Output the result
End

Figure 4: Flowchart of improved ACA used in UAV trajectory planning

5 Simulation experiments

Figure 5 describes a UAV's mission situation. The enemy's position size is 68km ×60 km. The rectangular block is target point, the dot is threat point coming from radars, missiles and so on. Its concrete position can be seen in Table 1. After entering the enemy side's defending region, UAV needs to complete the computation of trajectory optimizing according to the threaten environment.

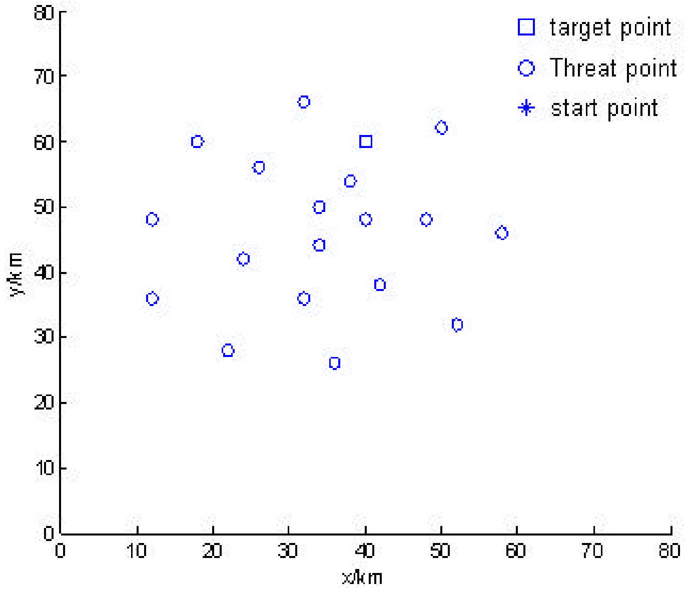

Figure 5: A UAV's task graph

Start point	(4,0)
Target point coordinate	(40,60)

Threat point coordinate and numeral order	1(18,60) 2(32,66.5) 3(51,62) 4(57.5,46)
	5(52,32) 6(36,26) 7(22,27) 8(12 ,36)
	9(12 ,48) 10(25.5,55) 11(48,48) 12(24,42)
	13(34,50) 14(38,53) 15(42,37) 16(40,48)
	17(34,43) 18(31,35.5

Table 1: Position of start point, target point, threat point

In order to find a feasible UAV flight trajectory, the paper divide the enemy's region into a 2 km ×2 km rectangular network. The point of intersection is the UAV's feasible trajectory node and its coordinates are definite. Optimize the route in order to determine the feasible node that the UAV passes. Adopt the above trajectory planning way based on ACA, and carry on the trajectory planning using the simulation technology. The basic ACA parameters were set to the following values: *a*= 1.0, ß= 2, ?= 0.3, *Q* = 100, *m* = 20, *Ncmax*=200. The improved ACA parameters were set to the following values: a=1.0, ß= 2, ?= 0. 9, *Q*=100, *m*=20, ?$_{min}$=0.01, *t*$_{min}$=0.01, *t*$_{max}$=10, *Ncmax*.200.

The optimal flight trajectories are shown in figure 6 and figure 7 obtained by using the basic ACA and the improved ACA.

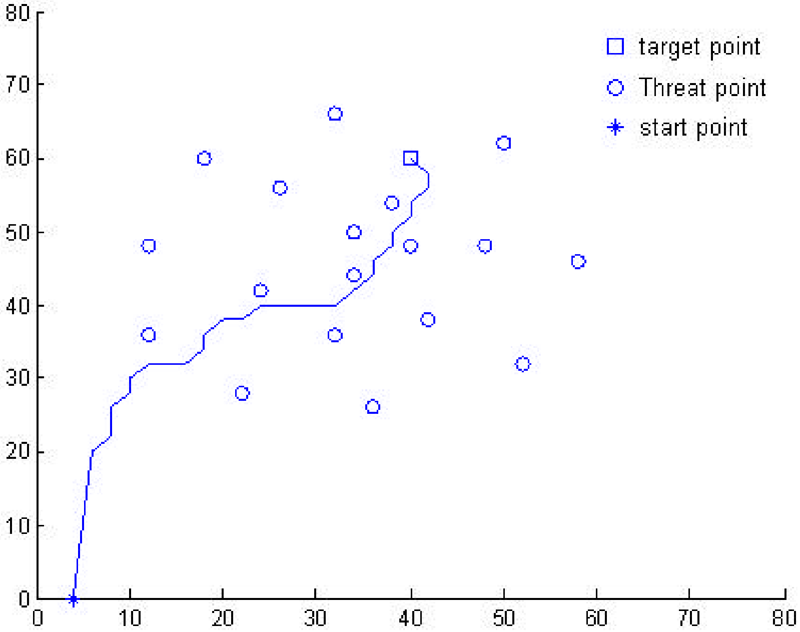

Figure 6: Optimal trajectory by the basic ACA

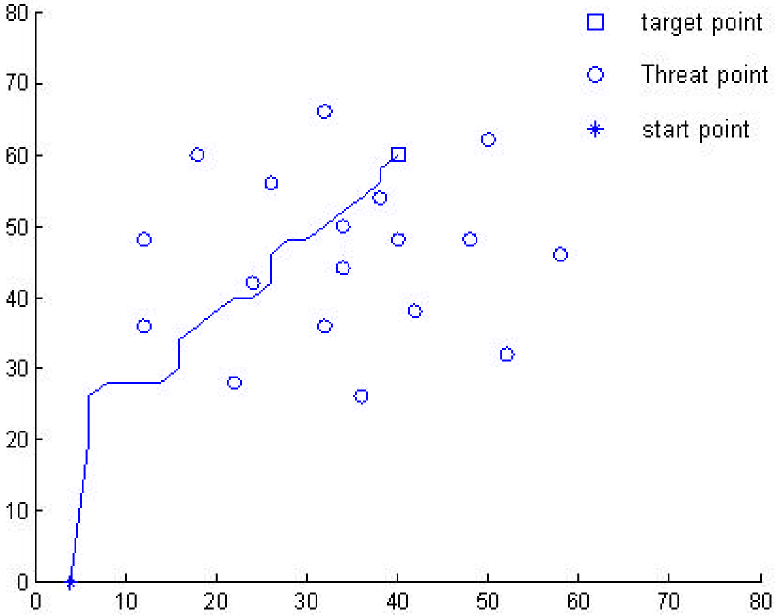

Figure 7: Optimal trajectory by the improved ACA

Contrasting the above two simulation results, we can see that, under the same threat condition, using the improved ACA can be possible to find a shorter trajectory and a more feasible UAV flight trajectory. This is because that the improved ACA has overcome the basic ACA's shortcomings which will fall into partially the most superior quickly, and has fully considered the integrity of trajectory planning, thus obtain the more optimal UAV flight trajectory.

However, by the improved ACA, we have found a minimal threat trajectory for UAV, which is fold line, not smooth line, to UAV it is not a feasible flight trajectory, so we should smooth the trajectory. After smoothing, the trajectory consists of line and arc, as shown in Figure 8 ~ Figure 10. Figure 11 gives the best solution of each iteration.

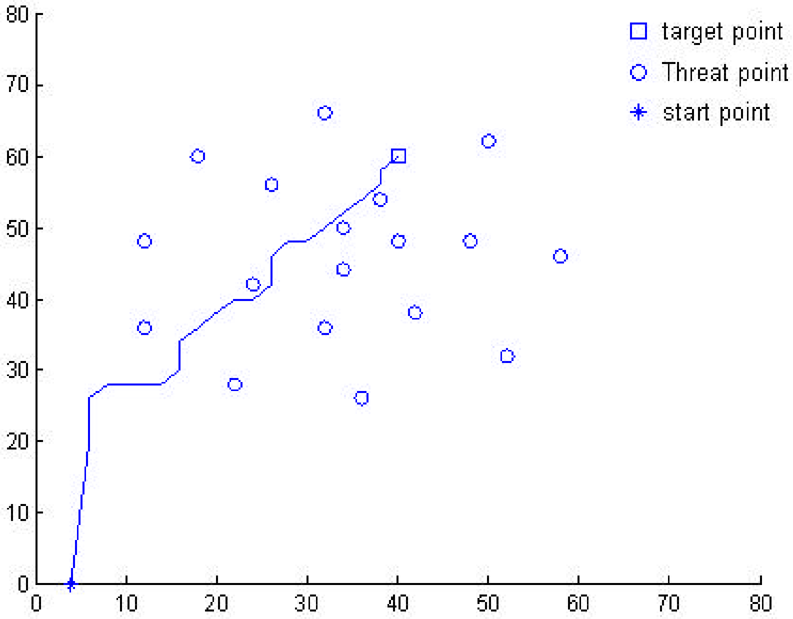

Figure 8: Optimal trajectory before smoothing

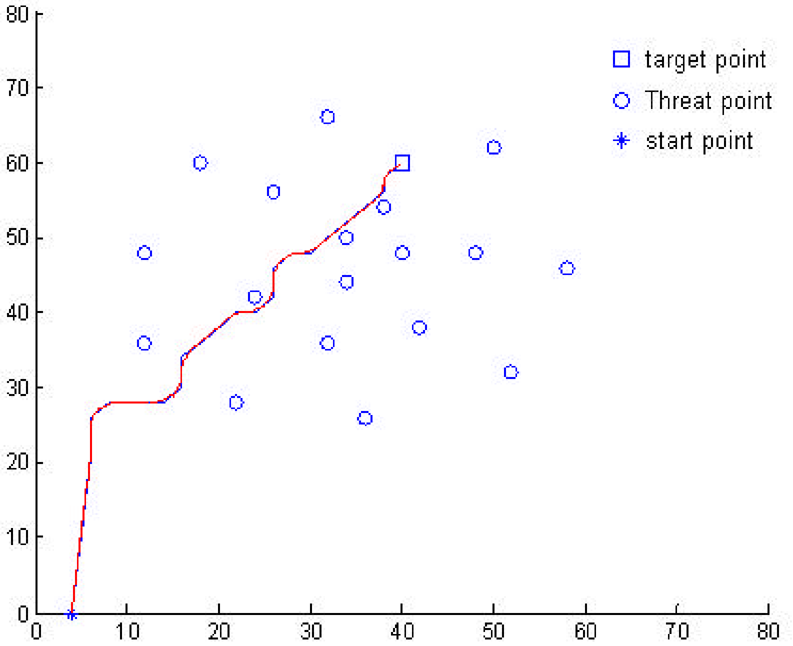

Figure 9: Optimal trajectory when smoothing

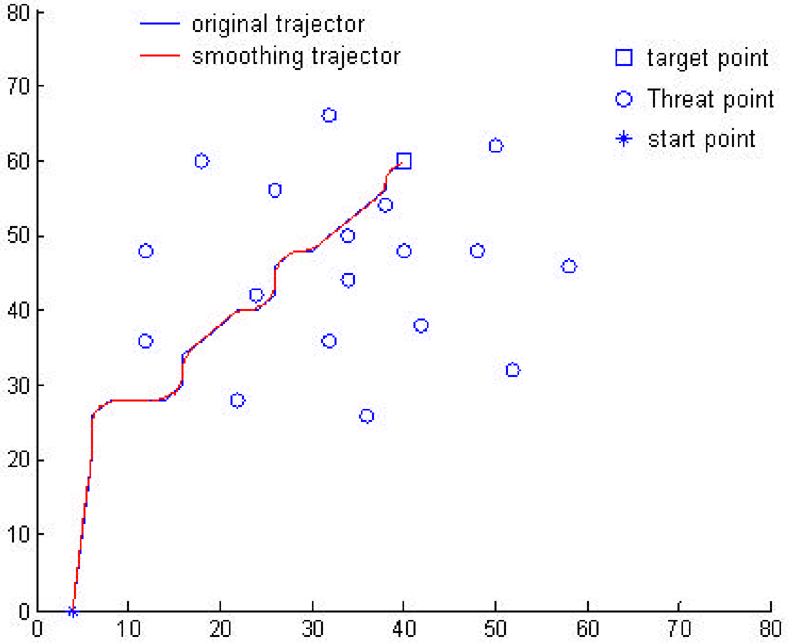

Figure 10: Experimental comparison of optimal trajectories

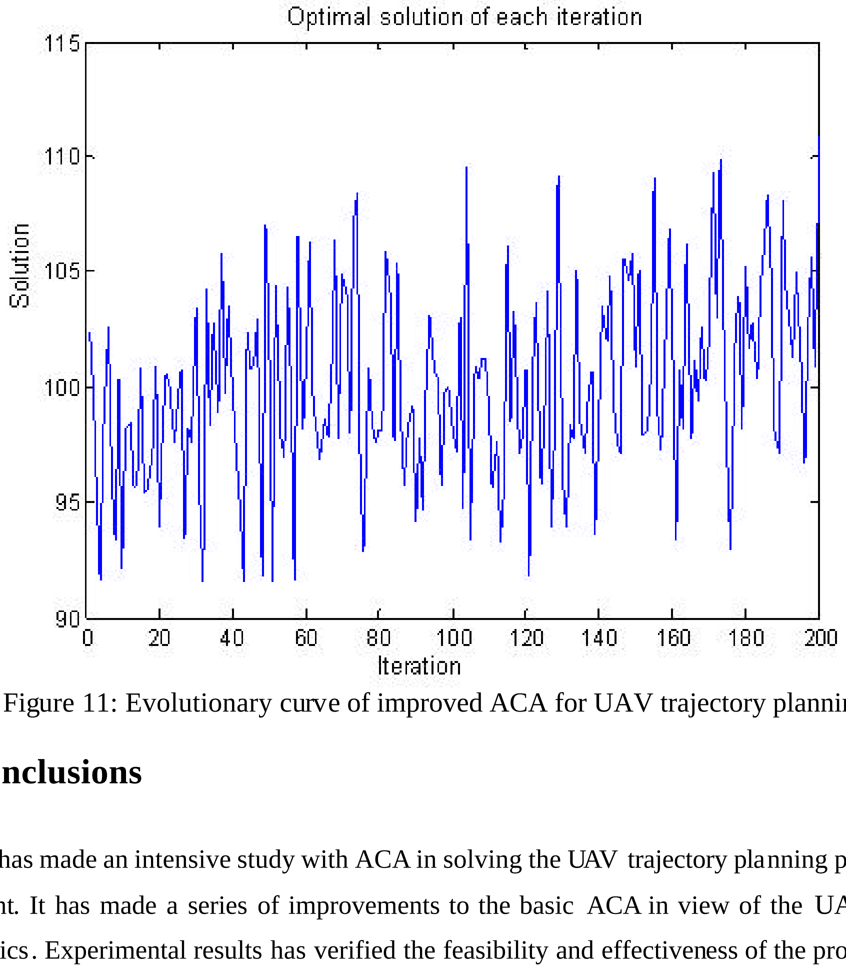

Figure 11: Evolutionary curve of improved ACA for UAV trajectory planning

6 Conclusions

This paper has made an intensive study with ACA in solving the UAV trajectory planning problem under complex environment. It has made a series of improvements to the basic ACA in view of the UAV trajectory planning characteristics. Experimental results has verified the feasibility and effectiveness of the proposed ACA in solving the UAV trajectory planning problem under complex environment. Future research will focuse on ACA for solving dynamic global optimal trajectory planning of UAV under complex environments.

7 Acknowledgement

The authors are grateful to the anonymous referees for their valuable comments and suggestions that have led to the better presentation of this paper. This work was supported by the Natural Science Foundation of China under grant #60604009, Aeronautical Science Foundation of China under grant #2006ZC51039 and "New Talent of Blue Sky" Foundation of Beihang University.

References

[1] Vant't Hart, S. A., Koemers, G.. J. M., Integration of sensor characteristics into unmanned aerial vehicle route planning, *Proceedings of 25th IEEE Digital Avionics Systems Conference*, 1-12, 2006

[2] Zelek, J.S., Dynamic path planning, *Proceedings of IEEE Conference on Systems, Man and Cybernetics*, 1285-1290, 1995.

[3] Pellazar, M. B., Vehicles route planning with constraints using genetic algorithms, *Proceedings of IEEE National Aerospace and Electronic Conference*, pp.392-399, 1998.

[4] Mclain, T., Beard, R., Trajectory planning for coordinated rendezvous of unmanned air vehicles, *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 4369-4372, 2000.

[5] Duan, H. B., Wang, D. B., Yu, X. F., Grid-based ACO algorithm for parameters tuning of NLPID controller and its application in flight simulator. *International Journal of Computational Methods*, 3(2):163-175, 2006.

[6] Duan, H. B., Ma, G. J., Wang, D. B., Yu, X. F., An improved ant colony algorithm for solving continuous space optimization problems. *Journal of System Simulation*, 19(5): 974-977, 2007.

[7] Duan, H. B., Yu, X. F., MA, G.J., Novel hybrid approach for fault diagnosis in 3-DOF flight simulator based on BP neural network and ant colony algorithms. *Proceedings of the IEEE Swarm Intelligence Symposium*, 371-374, 2007.

[8] Dorigo, M., Maniezzo, V., Colorni, A., Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1): 29-41, 1996.

[9] Duan, H.B., Wang D. B., ZHU J. Q., Huang X. H., New development on ant colony algorithm theory and its applications. *Control & Decision*, 19(12): 1321 -1326, 1340, 2004.

[10] Bonabeau, E., Yang, J. B., Inspiration for optimization from social insect behavior. *Nature*, 406(6): 39-42. 2000.

[11] Srolle, S., Rysdyk, R., Flight path following guidance for unmanned air vehicles with pan-tilt camera for target observation, *Proceedings of the IEEE Digital Avionics Systems Conference*,2:8.B.3 - 81-12, 2003.

[12] Yao, H.Q., Quan P., Jian, G.Y., Flight path planning of UAV based on heuristically search and genetic algorithms,*Proceedings of the IEEE 32nd Annual Conference*,45-50,2005.

[13] Liu, C.A., Li, W.J, Wang, H.P., Path planning for UAVs based on ant colony, Journal of the Air Force Engineering University, 2(5):9-12, 2004.

[14] Anderson, E.P., Beard, R.W., McLain, T.W., Real-time dynamic trajectory smoothing for unmanned air vehicles, *IEEE Transactions on Control Systems Technology*,13(3):471-477,2005.

[15] Dorigo, M., Gambardella, L. M., Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation*, 1(1): 53-66, 1997.

[16] Thomas, S., Holger H. H., Max-Min ant system, *Future Generation Computer Systems*, 16(8): 889-891, 2000.

[17] Wang, Y., Xie, J.Y., An adaptive ant colony optimization algorithm and simulation. *Journal of System Simulation*, 14(1): 31-33, 2002.