Aerospace Science and Technology 14 (2010) 535–541

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle (UCAV) path planning

Chunfang Xu, Haibin Duan*, Fang Liu

National Key Laboratory of Science and Technology on Holistic Flight Control, School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, PR China

ARTICLE INFO

Article history:
Received 30 March 2010
Accepted 19 April 2010
Available online 21 April 2010

Keywords:
Uninhabited Combat Air Vehicle (UCAV)
Artificial Bee Colony (ABC)
Chaos theory
Path planning

ABSTRACT

Path planning of Uninhabited Combat Air Vehicle (UCAV) is a rather complicated global optimum problem which is about seeking a superior flight route considering the different kinds of constrains under complex combat field environment. Artificial Bee Colony (ABC) algorithm is a new optimization method motivated by the intelligent behavior of honey bees. In this paper, we propose an improved ABC optimization algorithm based on chaos theory for solving the UCAV path planning in various combat field environments, and the implementation procedure of our proposed chaotic ABC approach is also described in detail. Series of experimental comparison results are presented to show the feasibility, effectiveness and robustness of our proposed method.

1. Introduction

Uninhabited Combat Aerial Vehicle (UCAV) is one of the inevitable trends of the modern aerial weapon equipments owing to its potential to perform dangerous, repetitive tasks in remote and hazardous environments [12]. Research on UCAV can directly affect battle effectiveness of the air force, therefore is crucial to safeness of a nation. Path planning is an imperative task required in the design of UCAV, which is to search out an optimal or near-optimal flight path between an initial location and the desired destination under specific constraint conditions. Series of algorithms have been proposed to solve this complicated optimization problem, including the A* algorithm, evolutionary computation [12], particle swarm optimization [2], genetic algorithm (GA) [9] and ant colony algorithm [11]. However, those methods can be easily trapped into the local best, hence would probably end up without finding a satisfying path. In our paper, we mainly focus on UCAV path planning in two dimensions.

Artificial Bee Colony (ABC) algorithm was originally presented by Dervis Karaboga in 2007 [5], under the inspiration of collective behavior on honey bees, and it has been proved to possess a better performance in function optimization problem, compared with genetic algorithm, differential evolution (DE) algorithm and particle swarm optimization (PSO) algorithm [5,6]. As we know, usual optimization algorithms conduct only one search operation in one iteration, for example the PSO algorithm carries out global search at the beginning and local search in the later stage. Compared with the usual algorithms, the major advantage of ABC algorithm lies in that it conducts both global search and local search in each iteration, and as a result the probability of finding the optimal parameters is significantly increased, which efficiently avoid local optimum to a large extent. Although the ABC algorithm has rarely been used in path planning field before, yet due to the above advantages we described, we adopted this algorithm to figure out the flight path. What is more, considering the outstanding performance of chaos theory in jumping out of stagnation, we introduced it to improve the robustness of basic ABC algorithm, and the comparative experimental results testified that our proposed method manifests better performance than the original ABC algorithm.

The remainder of this paper is organized as follows. Section 2 introduces the threat resource and objective function in UCAV path planning. Section 3 described the principle of basic ABC algorithm, while Section 4 specified implementation procedure of our proposed chaotic ABC algorithm. Then, in Section 5, series of comparison experiments are conducted. Our concluding remarks are contained in the final section.

2. Environmental modeling for UCAV path planning

2.1. Threat resource model in UCAV path planning

Modeling of the threat sources is the key task in UCAV optimal path planning. In our model, define the starting point as S and the target point as T, as is shown in Fig. 1. There are some threatening areas in the task region, such as radars, missiles, and artillery, which all are presented in the form of a circle, inside of which will be vulnerable to the threat with a certain probability proportional to the distance away from the threat center, while out of which

* Corresponding author. Tel.: +86 10 8231 7318.
E-mail address: hbduan@buaa.edu.cn (H.B. Duan).

1270-9638/$ – see front matter
doi:10.1016/j.ast.2010.04.008

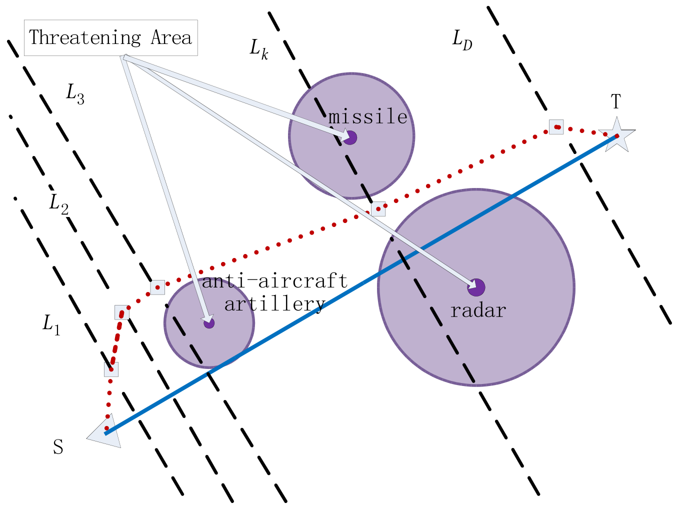


Fig. 1. Typical UCAV battle field model.

will not be attacked. The flight task is to generate an optimal path between S and T considering all these threatening areas.

First connect point S and point T, then divide segment ST into $(D+1)$ equal portions. At each segment point, draw the vertical line of ST, denoted as $L_1, L_2, \ldots, L_k, \ldots, L_D$. Take a discrete point at each vertical segment L_k, engendering a collection of discrete points $C = \{S, L_1(x(1), y(1)), L_2(x(2), y(2)), \ldots, L_k(x(k), y(k)), \ldots, L_D(x(D), y(D)), T\}$, and connect them in sequence to form a path. In this way, the path planning problem is turning into optimizing the coordinates series to achieve a superior fitness value of the objective function.

To accelerate the search speed of the algorithm, we can let line ST be the x axis and take the coordinate transformation on each discrete point $(x(k), y(k))$ according to formula (1), where θ is the angle that the original x axis contrarotates to parallel segment ST, while (x_s, y_s) represents the coordinates in the original coordinate system.

$$\begin{bmatrix} x'(k) \\ y'(k) \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x(k) - x_s \\ y(k) - y_s \end{bmatrix} \tag{1}$$

Thus, the x coordinate of each point can be obtained by a simple formula $x'(k) = \frac{|ST|}{D+1} \cdot k$, therefore the collection C of points can be simplified into $C' = \{0, L_1(y'(1)), L_2(y'(2)), \ldots, L_k(y'(k)), \ldots, L_D(y'(D)), 0\}$, which can greatly reduce the computational cost.

2.2. The performance evaluation function of route optimization

The performance indicators of the UCAV route mainly include the threat cost J_t and the fuel cost J_f, the calculating formulas of which are presented as follows:

$$J_t = \int_0^L w_t \, dl \tag{2}$$

$$J_f = \int_0^L w_f \, dl \tag{3}$$

where w_t and w_f are variables close related with the current path point and changing along with 'l', which respectively present the threat cost and fuel cost of each line segment on the route, while L is the total length of the generated path.

In order to simplify the calculations, a computationally more efficient and acceptably accurate approximation to the exact solution is adopted. In this work, the threat cost of each edge connecting two discrete points was calculated at five points along it, as is shown in Fig. 2.

If the ith edge is within the effect range, the threat cost is given by the expression [7,1]:

$$w_{t,L_i} = \frac{L_i}{5} \cdot \sum_{k=1}^{N_t} t_k \cdot \left(\frac{1}{d_{0.1,i,k}^4} + \frac{1}{d_{0.3,i,k}^4} + \frac{1}{d_{0.5,i,k}^4} + \frac{1}{d_{0.7,i,k}^4} + \frac{1}{d_{0.9,i,k}^4} \right) \tag{4}$$

where N_t is the number of threatening areas, L_i is the ith sub-path length, $d_{0.1,i,k}$ is the distance from the $1/10$ point on the ith edge to the kth threat, and t_k is the threat level of kth threat.

Consuming that the speed of UCAV is a constant, then the fuel cost of the path J_f can be considered equal to L, the total length of path.

The total cost for traveling along the trajectory comes from a weighted sum of the threat and fuel costs, as is defined in formula (5),

$$J = kJ_t + (1-k)J_f \tag{5}$$

where k is a variable between 0 and 1 (0.5 in our algorithm), which gives the designer certain flexibility to dispose relations between the threat exposition degree and the fuel consumption. When k is more approaching 1, a shorter path is needed to be planned, and less attention is paid to the radar's exposed threat. Otherwise, when k is more approaching 0, it requires avoiding the threat as far as possible on the cost of sacrifice the trajectory length. The optimized path is founded only when function J reaches its minimal value.

3. Principles of the basic ABC algorithm

Karlvon Frisch, a famous Nobel Prize winner, found that in nature, although each bee only performs one single task, yet through a variety of information communication ways between bees such as waggle dance and special odor, the entire colony can always easily find food resources that produce relative high amount nectar, hence realize its self-organizing behavior [4].

In order to introduce the self-organization model of forage selection that leads to the emergence of collective intelligence of honey bee swarms, first, we need to define three essential components: food sources, unemployed foragers and employed foragers.

(1) *Food sources* (A and B in Fig. 3)
For the sake of simplicity, the "profitability" of a food source can be represented with a single quantity. In UCAV path planning problem, the position of a food source represents a possible parameter solution to the optimization problem and the nectar amount of a food source corresponds to the similarity value of the associated solution.

(2) *Unemployed foragers*
If it is assumed that a bee has no knowledge about the food sources in the search field, bee initializes its search as an unemployed forager [3]. Unemployed foragers are continually at look out for a food source to exploit. There are two types of unemployed foragers: scouts and onlookers.
- *Scouts* (S in Fig. 3): If the bee starts searching spontaneously for new food sources without any knowledge, it will be a scout bee.
- *Onlookers* (R in Fig. 3): The onlookers wait in the nest and search the food source through sharing information of the employed foragers, and there is a greater probability of onlookers choosing more profitable sources.

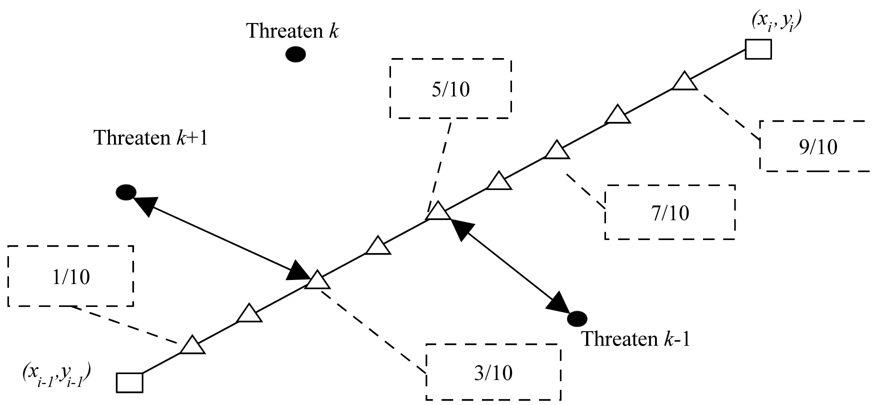


Fig. 2. Computation of threat cost.

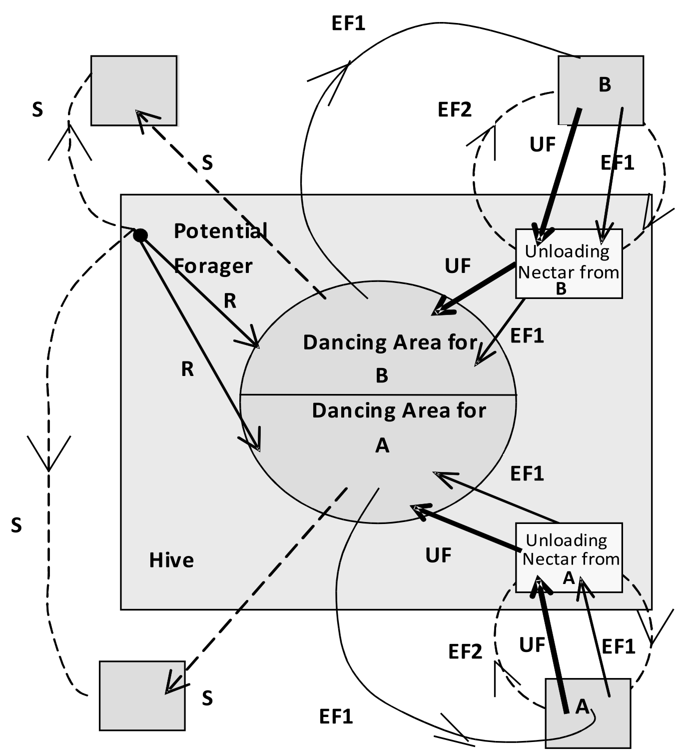


Fig. 3. The behavior of honey bee foraging for nectar.

(3) *Employed foragers*
They are associated with a particular food source which they are currently exploiting. They carry with them information about this particular source, the profitability of the source and share this information with a certain probability. After the employed foraging bee loads a portion of nectar from the food source, it returns to the hive and unloads the nectar to the food area in the hive. There are three possible options related to residual amount of nectar for the foraging bee.

- If the nectar amount decreased to a low level or exhausted, foraging bee abandons the food source and become an unemployed bee (UF in Fig. 3).
- If there are still sufficient amount of nectar in the food source, it can continue to forage without sharing the food source information with the nest mates (EF2 in Fig. 3).
- Or it can go to the dance area to perform waggle dance for informing the nest mates about the food source (EF1 in Fig. 3).

In this way, the bees finally can construct a relative good solution of the multimodal optimization problems.

At the initial moment, all the bees without any prior knowledge play the role of detecting bees. After a random search for bee sources, the detecting bees can convert into any kind of bees above in accordance with the profit of the searched food sources. The changing rules are described as follows:

When the profit of the food source the bee searched is higher than the threshold, it becomes a leading bee, goes on exploring nectar and also recruits more bees (EF1) to explore together. When the profit of related food source is relative low, it gives up the food source, and again becomes a detecting bee to search for a new food source (UF). When the profit is less than certain threshold, it follows leading bees to explore nectar. When searching times around hive exceed a certain limit but still could not find a good resource, abandon the source and find a new one.

4. Principles of the chaotic ABC algorithm

4.1. Introduction to chaos theory

Chaos is the highly unstable motion of deterministic systems in finite phase space which often exists in nonlinear systems. Chaos theory is epitomized by the so-called 'butterfly effect' detailed by Lorenz [8]. Attempting to simulate numerically a global weather system, Lorenz discovered that minute changes in initial conditions steered subsequent simulations towards radically different final stales, rendering long-term prediction impossible in general. Until now, chaotic behavior has already been observed in the laboratory in a variety of systems including electrical circuits, lasers, oscillating chemical reactions, fluid dynamics, as well as computer models of chaotic processes. Chaos theory has been applied to a number of fields, among which one of the most applications was in ecology, where dynamical systems have been used to show how population growth under density dependence can lead to chaotic dynamics.

Sensitive dependence on initial conditions is not only observed in complex systems, but even in the simplest logistic equation. In the well-known logistic equation:

$$x_{n+1} = 4x_n(1 - x_n) \tag{6}$$

where $0 < x_n < 1$, a very small difference in the initial value of x would give rise to large difference in its long-time behavior, which is the basic characteristic of chaos. The track of chaotic variable can travel ergodically over the whole space of interest. The variation of the chaotic variable has a delicate inherent rule in spite of the fact that its variation looks like in disorder. Therefore, after each search round, we can conduct the chaotic search in the neighborhood of the current optimal parameters by listing a certain number of new generated parameters through chaotic process.

In this way, we can make use of the ergodicity and irregularity of the chaotic variable to help the algorithm to jump out of the local optimum as well as finding the optimal parameters. The experimental results in Section 5 show the efficiency of our algorithm.

4.2. Chaotic ABC approach for solving the path planning problem

Due to the flexibility, versatility and robustness in solving optimization problems, ABC algorithm has already aroused intense interest. However, there still exist some flaws on this algorithm, such as the large number of iterations to reach the global optimal solution and the tendency to converge prematurely. In order to overcome these flaws of ABC and upon the merits of chaotic variable, chaotic ABC algorithm, which integrates ABC with chaotic variable, was proposed in our work. After the search process of each bee, conduct the chaotic search in the neighborhood of current best solution in order to choose one better solution into next generation. In this way, our proposed algorithm takes the advantage of the characteristics of the chaotic variable to make the individuals of subgenerations distributed ergodically in the defined space and thus to avoid from the premature of the individuals, as well as to increase the speed of reaching the optimal solution.

The implementation procedure of our proposed chaotic ABC approach to UCAV path planning can be described as follows:

Step 1: According to the environmental modeling in Section 2, initialize the detailed information about the path planning task, as well as the threaten information including the coordinates of threat centers, threat radiuses and threat levels. In order to simplify the calculation, conduct the coordinate transformation on discrete points related with the task according to formula (1).

Step 2: Initialize the parameters of artificial bee colony optimization algorithm, such as the population of the bee colony N_s, the number of employed bees N_e and the number of the unemployed bees N_u, which satisfy the condition shown as follows:

$$N_s = N_e + N_u \tag{7}$$

Obviously, a larger N_s will contribute to a larger possibility of finding the best solution of the problem, however, it also means an increased computing complexity of the algorithm. In general, we define $N_e = N_u$, and according to our special problem, we set $N_s = 60$. Denote the largest searching times with *Limit* (30 in our experiments), current iterations with T, and the largest iterations with $T_{\max}$. Initialize within the bound of the battlefield the employed bee population of D-dimensional parameters $C' = \{y'(1), y'(2), \ldots, y'(k), \ldots, y'(D)\}$, which represent the y coordinates of each discrete point as we discussed in Section 2, while the corresponding x coordinates could be easily obtained by the formula $x'(k) = \frac{|ST|}{D+1} \cdot k$. Each group of parameters can engender a path that leading the UAV from the starting point S to the target point T, and the goal is to find the optimal combination of parameters that can provide relative satisfactory performance. Initialize the search time of each bee $Bas = 0$, and the starting iteration $T = 1$.

Step 3: According to the parameters of the employed bees, calculate the cost of each path formed by relative parameters based on formulas (2)–(5). The smaller the cost value is, the better performance the path maintains.

Step 4: The employed bees search around their current positions (parameters) to find new solutions, and update their positions if the new cost value is lower than the original value.

The search strategy can be described as follows: for the ith employed bee, first engender a random integer j between 1 and D and a random integer k between 1 and N_e, then the jth parameter of the ith employed bee could be updated by formula:

$$y_i''(j) = y_i'(j) + \left(y_i'(j) - y_k'(j)\right) \cdot (rand - 0.5) \cdot 2 \tag{8}$$

where *rand* represents a random value between 0 and 1. Calculate the new cost value of the updated parameters and choose the one that possesses a lower cost as the new employed bee.

Step 5: The unemployed bees apply the roulette selection method to choose the bee individual that maintains a relatively low cost value as the leading bee according to the calculated cost results of employed bees. Each recruited unemployed bee continues to search new solutions just around the leading bee's solution space similar with Step 4, and calculate their cost values. If the capability of the new solution is better than the original one, the unemployed bee converts into an employed bee, which means that update the positions of the employed bees, and continue exploring with *Bas* re-initialized as 0, or else, keep searching around, and its *Bas* value plus one.

Step 6: If the search times *Bas* is larger than certain threshold *Limit*, the employed bee gives up the solution, and re-search the new food resources, which is realized by re-initializing the parameters and calculating the cost value.

Step 7: Store the best solution parameters and the best cost value.

Step 8: Conduct the chaotic search around the best solution parameters based on formula (6) after transforming the parameters ranges into $(0, 1)$. Among the engendered series of solutions, select the best one and use it to replace a random employed bee.

Step 9: If $T < T_{\max}$, go to Step 4. Otherwise, output the optimal parameters and optimal cost value.

The detailed procedure can also be shown with Fig. 4.

4.3. Complexity analysis of the chaotic ABC algorithm

From the description of the chaotic ABC algorithm, it is clear that the computational complexity of the algorithm is $O(D^2 T_c N_s)$ [10], where D represents the dimension of the problem to be solved, T_c denotes the iterations required to obtain the optimized solution, and N_s is the number of the bee population. The ABC algorithm can get the optimized solution with the desired computational cost. Compared with standard ABC algorithm, our proposed chaotic ABC algorithm calculates only $T_c N_c$ times more, where N_c represents the number of chaotic search process in each iteration. Since in our algorithm, N_c is small while D is a relatively large value, the computational complexity of the chaotic ABC algorithm is essentially equal to that of the standard ABC algorithm.

5. Experimental results

In order to investigate the feasibility and effectiveness of the proposed method in this work, series of experiments are conducted, and further comparative experimental results with the standard ABC algorithm are also given.

Set the coordinates of the starting point as $(11, 11)$, and the target point as $(75, 75)$, while the initial parameters of ABC algorithm were set as: $N_s = 60$, $N_e = 30$, $N_u = 30$, $T_{\max} = 100$, $Limit = 30$.

Respectively assume D as 10, 20 and 30 to carry our experiments, the results of which are shown in Figs. 5–9. Fig. 5 is the path planning result of chaotic ABC algorithm when D is set as 10, and the achieved path shown in the figure obviously maintains a favorable performance, hence proves the feasibility of our algorithm without any doubt.

When $D = 10$, the experimental results of standard ABC and chaotic ABC have slightly differences due to the calculating complexity. However, when the value of D is increased to 20, even to 30, we can clearly see the superiority of our proposed method

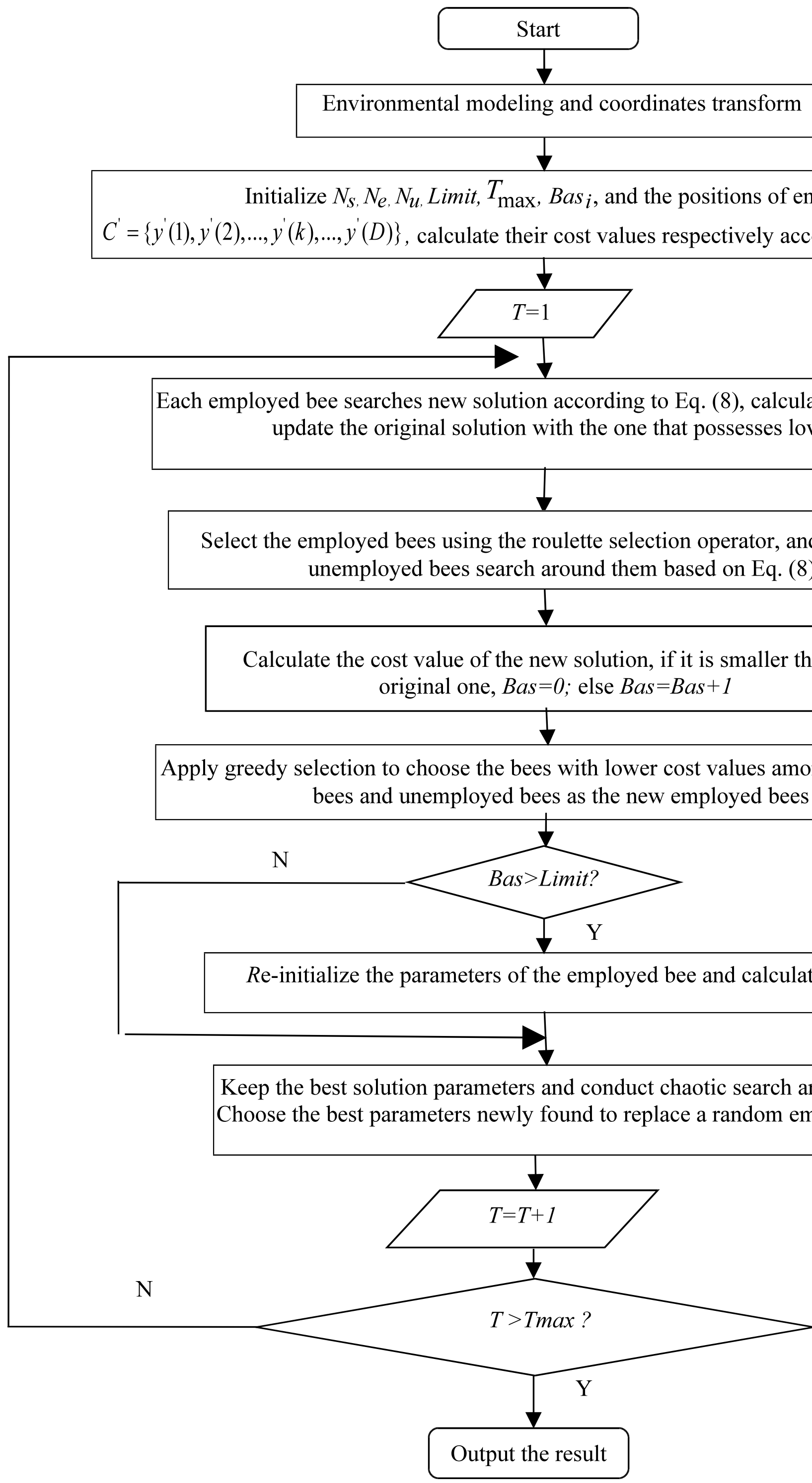


Fig. 4. The procedure of our proposed method.

over the standard ABC algorithm in the comparative experimental results shown in Figs. 6–9.

To further prove the performance of our proposed method against standard ABC algorithm, we run the program for 100 times to obtain the average cost of our generated best path. It turns out that the average cost of our algorithm is 50.9346, while the average cost of standard ABC algorithm is 53.9071, apparently showing that our method can find the feasible and optimal path for UCAV more stable than basic ABC algorithm, and can effectively solve the path planning problem of UCAV in complex combat field environment.

When the threats move, we can recalculate the path according to current threat positions. The simulation results can be shown in Fig. 10, which shows the feasibility of chaotic ABC algorithm under moving threatens.

From the above experimental results, we can clearly see that using standard ABC algorithm could possibly lead to a path that does not satisfy the requirements, especially when the optimized

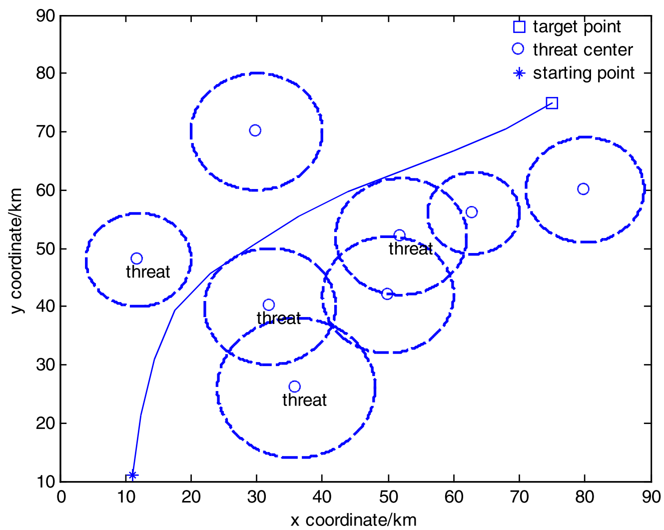


Fig. 5. The path planning result using chaotic ABC algorithm, $D = 10$.

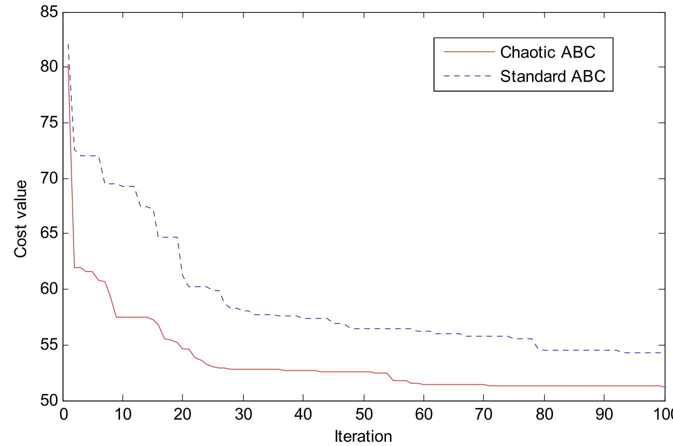
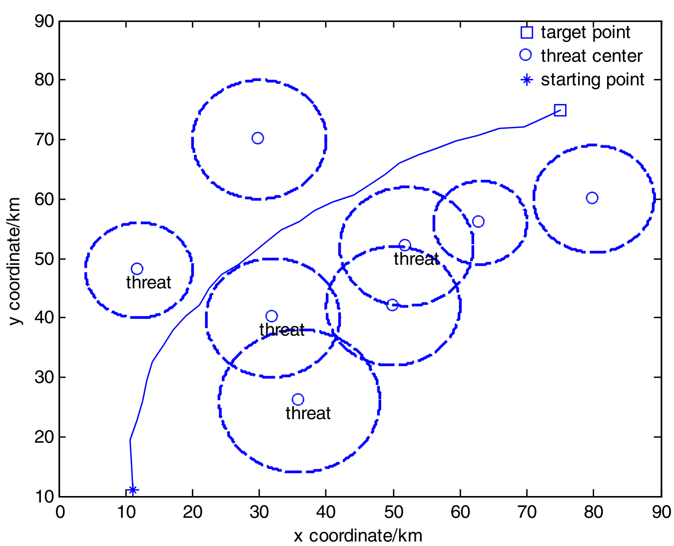
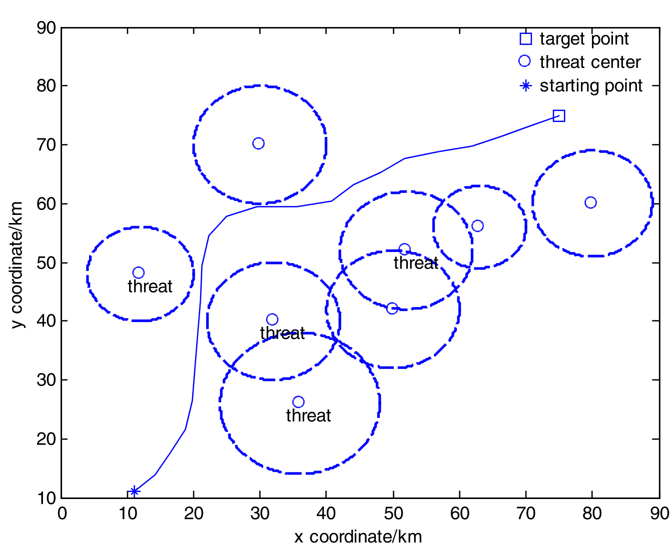


Fig. 7. The evolution curves of two algorithms, $D = 20$.

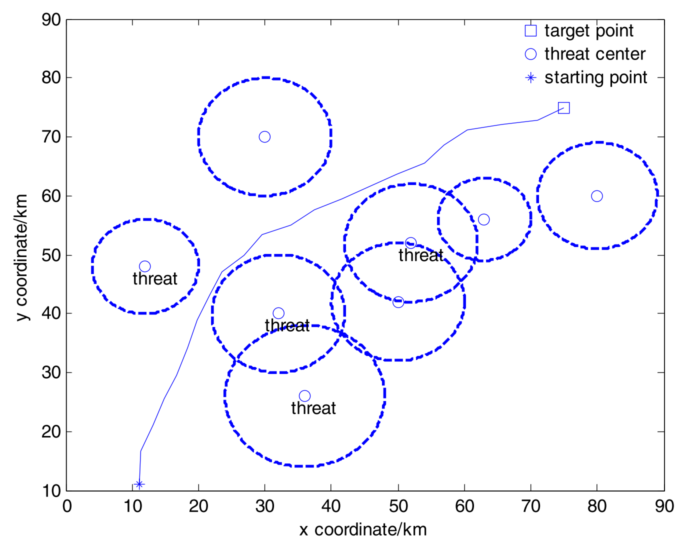


(a) Chaotic ABC algorithm

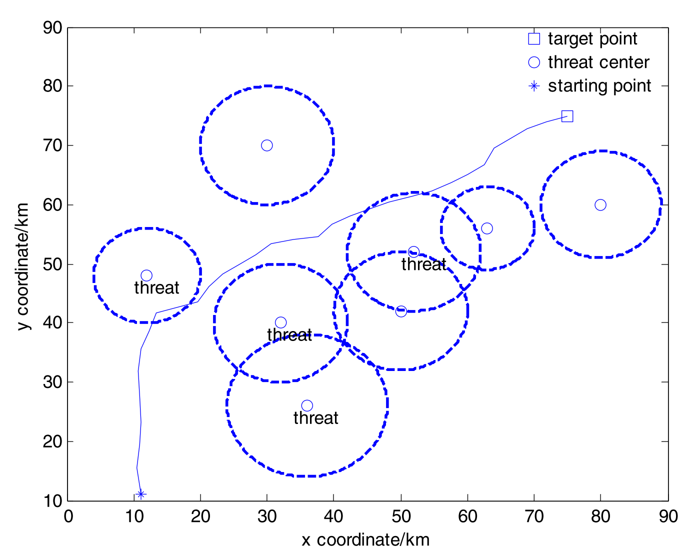


(b) Standard ABC algorithm

Fig. 6. The comparative path planning results, $D = 20$.



(a) Chaotic ABC algorithm



(b) Standard ABC algorithm

Fig. 8. The comparative path planning results, $D = 30$.

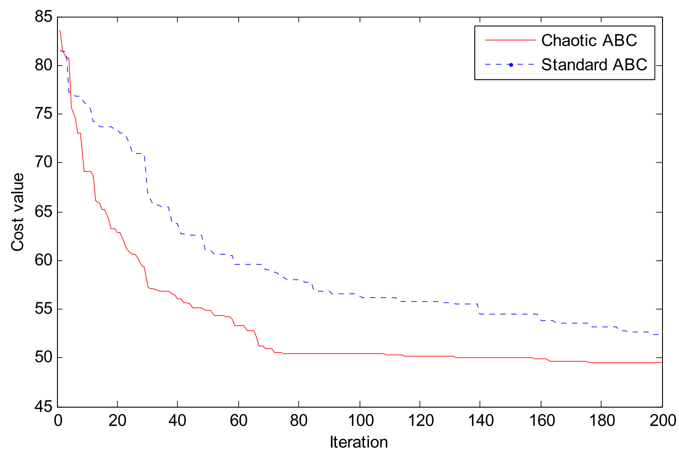
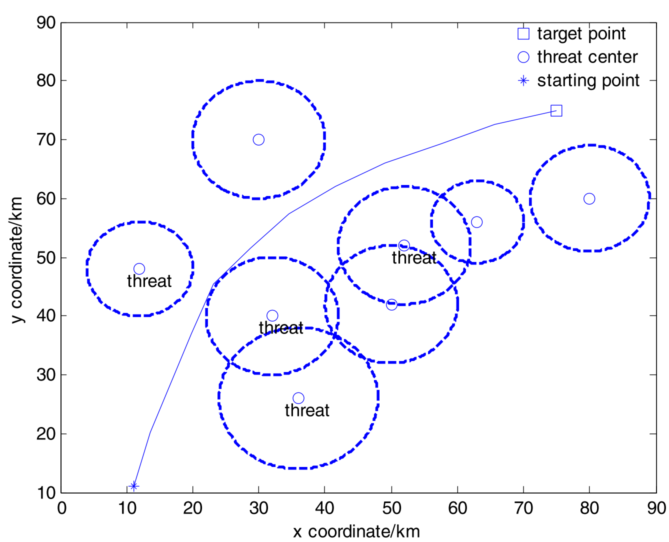
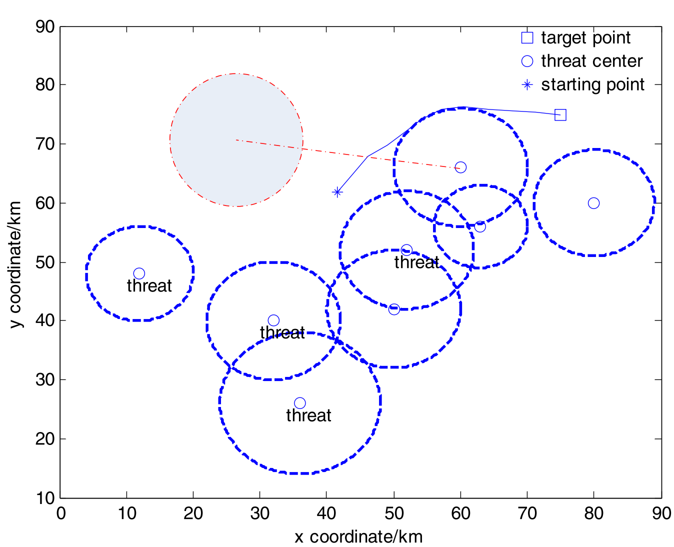


Fig. 9. The evolution curves of two algorithms, $D = 30$.



(a) Initial optimized path



(b) Recalculated path after the threaten moves

Fig. 10. Experimental results on moving threatens.

dimension increases. Therefore, we make use of the ergodicity of chaotic variable to help the basic ABC algorithm to jump out of the local best and obtain a favorable path.

6. Concluding remarks

This paper presents a novel chaotic ABC approach for UCAV path planning problem in complicated combat field environment. Utilizing the ergodicity and irregularity of the chaotic variable to help the basic ABC algorithm to jump out of the local optimum as well as speeding up the process of finding the optimal parameters. The simulation experiments show that our proposed method is a feasible and effective way in UCAV path planning, and it is also flexible, in that dynamic environments and pop-up threats are easily incorporated. The experimental comparison also shows the stability and superiority of our method over the standard ABC algorithm, which provides a more effective way for UCAV path planning.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 60975072 and 60604009), the Program for New Century Excellent Talents in University of China (Grant No. NCET-10-0021), Aeronautical Science Foundation of China, and the Beijing NOVA Program Foundation of China (Grant No. 2007A0017), Graduate Innovation Practice Foundation of Beihang University, China, and the Fundamental Research Funds for the Central Universities (Grant No. YWF-10-01-A18).

References

[1] R. Daniel, P.E. Andries, Social networks in simulated multi-robot environment, International Journal of Intelligent Computing and Cybernetics 1 (1) (2008) 110–127.
[2] H.B. Duan, G.J. Ma, D.L. Luo, Optimal formation reconfiguration control of multiple UCAVs using improved particle swarm optimization, Journal of Bionic Engineering 5 (4) (2008) 215–223.
[3] H.B. Duan, C.F. Xu, Z.H. Xing, Artificial bee colony optimization based quantum evolutionary for continuous optimization problems, International Journal of Neural Systems 20 (1) (2010) 39–50.
[4] M. Fathian, B. Amiri, A. Maroosi, Application of honey bee mating optimization algorithm on clustering, Applied Mathematics and Computation 190 (2) (2007) 1502–1513.
[5] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of Global Optimization 39 (3) (2007) 459–471.
[6] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing 8 (1) (2008) 687–697.
[7] C.A. Liu, W.J. Li, H.P. Wang, Path planning for UAVs based on ant colony, Journal of the Air Force Engineering University 2 (5) (2004) 9–12.
[8] E.N. Lorenz, Deterministic nonperiodic flow, Journal of the Atmospheric Sciences 20 (3) (1963) 130–141.
[9] Y.X. Wang, Z.J. Chen, Genetic algorithms (GA) based flight path planning with constraints, Journal of Beijing University of Aeronautics and Astronautics 25 (3) (1999) 355–358.
[10] C.F. Xu, H.B. Duan, Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft, Pattern Recognition Letters (2009), doi:10.1016/j.patrec.2009.11.018.
[11] W. Ye, D.W. Ma, H.D. Fan, Algorithm for low altitude penetration aircraft path planning with improved ant colony algorithm, Chinese Journal of Aeronautics 18 (4) (2005) 304–309.
[12] C.W. Zheng, L. Li, F.J. Xu, Evolutionary route planner for unmanned air vehicles, IEEE Transactions on Robotics and Automation 21 (4) (2005) 609–620.