2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP)
June 9 – 11, 2013, Beijing, China

Unmanned Combat Aerial Vehicles Path Planning Using a Novel Probability Density Model Based on Artificial Bee Colony Algorithm

Bai Li, Ligang Gong, and Chunhui Zhao

Abstract—**Path planning of unmanned combat aerial vehicle (UCAV) aims to seek an optimal flight route considering threats and constraints along the way towards the terminal target. This paper proposed a novel probability density model to transform the initial path planning task into a numerical problem, which shows higher accuracy in comparison with the traditional circle treat model. The well-known Artificial Bee Colony algorithm (ABC) is used to settle this corresponding optimization problem and comparisons are made between the proposed algorithm and other intelligence algorithms regarding convergence rate and efficiency in various series of combat fields. Experimental results verified with statistical significance the superiority of ABC for the UCAV path planning problem.**

I. Introduction

IN recent years, unmanned combat aerial vehicle (UCAV) has been of high interest to many military organizations throughout the world, considering its ability to work in dangerous or complicated environments [1]–[3]. Path planning system is a crucial component of autonomous control module in UCAV, which provides an optimal path from the starting point to the desired destination. During the flight, artificial threats and some natural constraints are supposed to avoid.

For a UCAV path planning task, the optimal solution corresponds to one that minimizes the traveled distance, average altitude, consumption of fuel, exposure to radar or artillery, etc [4]. As guarding and defending weapons develop, the complexity of modeling these artificial threats significantly grows. To cope with the increasing complexity, researchers have gradually shifted their interest away from deterministic algorithms. To avoid an inefficient enumerating process, intelligence algorithms have been developed and investigated in recent years, such as genetic algorithm (GA) [5], differential evolution algorithm (DE) [6], particle swarm optimization algorithm (PSO) [7] as well as the recently proposed artificial bee colony algorithm (ABC) [8]. PSO is inspired by the social behavior of bird flocking, where a swarm of particles move in the search space for appropriate solutions and every particle owns a position vector as well as a velocity vector. Each particle records its own best position so far and a global current best position is readily available for adjustments in vectors of particles. ABC was firstly proposed by Dervis Karaboga in 2005, which imitated the foraging behavior of bee swarms. In this algorithm, both local exploitation and global exploration are conducted in each iteration. It works well in exploration and has aroused great concern since then.

Before the adoption of these algorithms, an appropriate mathematical model needs to be built, which transfers the path planning task into a numerical optimization problem. It is noted that, as in previous work, the threat zone around a threat point has always been described by the traditional circle model, where researchers seldom considered the differences inside or outside the circle [9]–[11].

For the specific problem concerned in this paper, various algorithms such as GA [12], Immune GA (I-GA) [13], PSO [14], Quantum-behaved PSO (Q-PSO) [15], Master-slave parallel vector-evaluated GA (MPV-GA) [16], and Chaotic ABC (C-ABC) [9] have been developed.

Viewing previous work, we find that PSO defects to avoid getting trapped in local optimums. On the other hand, the true potentiality of ABC was seldom carefully considered for such real-time problems, and few rigorous experimental comparisons were made among the performances of these intelligence algorithms.

In this paper, a novel probability density model is introduced to differentiate different locations by their distances towards the threat center. A clear insight into the capability of ABC is also provided for this UCAV path planning problem. Simulation results verify, with statistical significance, paths obtained by ABC are superior to those by PSO. It is noted that this work preliminarily focuses on path planning in two dimensions.

The rest of this paper is organized as follows. In Section II, principle of the proposed environmental model is introduced. The procedure of ABC is elaborated in Section III. Several well-designed comparable simulation results are presented and carefully discussed in Section IV. And then a final conclusion is drawn in the end.

II. Model of UCAV Path Planning

For this specific UCAV path planning problem, researchers have been pursuing models that well describe the true environment, which account for artificial threats and natural constraints. In this work, the famous 2D model in [9] is basically used, and the probability density circle model

Manuscript received March 1, 2013. This work was sponsored in part by 5th and 6th National College Students' Innovative and Entrepreneurial Training Programs and supported by the Department of Advanced Engineering in Beihang University.

Bai Li is with the School of Advanced Engineering, Beihang University, Beijing, 100191, China (E-mail: libai@asee.buaa.edu.cn).

Ligang Gong is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, China (E-mail: glgbh@aspe.buaa.edu.cn).

Chunhui Zhao is with the State Key Laboratory of Industrial Control Technology, Department of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China (Tel: +86-13588312064; E-mail: chhzhao@zju.edu.cn).

978-1-4673-6249-8/13/$31.00 ©2013 IEEE

replaces the original circle model to characterize threat zones.

As in Fig. 1, a path that links up starting point and terminal destination is needed to obtain. At first, segment ST which connecting the starting and terminal points is drawn. Then it is divided into $(D+1)$ equal portions by D vertical lines which intersect ST at each segment point. These D lines are taken as new axis, onto which a series of points are set and connected one by one to form a path.

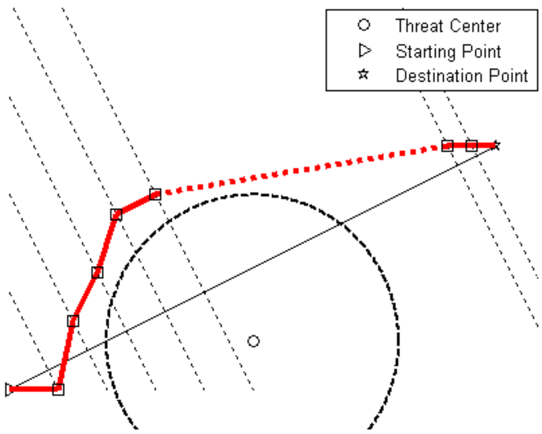

Fig. 1. The schematic diagram for combat field modeling. Note that all nodes are linked up to form a feasible path. During the flight, all threats are supposed to avoid all these known threat centers.

With the cost function to evaluate the quality of each path, traditional circle model has always been adopted to describe a threat zone from its center. In [9], positions outside a circle are believed to be indifferent because they are outside the boundary of a threat center. With respect to [17], the cost function is written as follows,

$$C_{\text{danger zones}} = L_{\text{inside } d,z} / \sum_{i=1}^{N} d_i \tag{1}$$

where n refers to the total number of danger zones in this district, $L_{\text{inside } d,z}$ refers to the total length of the subsections of the path which go through danger zones. Such circle threat model regards points inside a threat circle as indifferent.

But in a common sense, it is believed that there should not be a particular boundary outside of which the damage risk remains zero. For improvement, a probability density model is proposed as follows,

$$C_{\text{threat influence}} = \exp\left(-\frac{\sum_{i=1}^{n} \|d_i\|}{\delta}\right) \tag{2}$$

where $\|d_i\|$ denotes the distance from i-th threat center, the denominator δ is a parameter to control the shape of density function. For simplicity, it is assumed that all artificial threats or natural constraints can be expressed in this form but vary in their parameters. The cost function by probability density model is plotted under the condition of $\delta = 10$ and $n = 1$ as shown in Fig. 2. As the distance $\|d\|$ increases, the probability of suffering a risk gradually decreases but it would never be zero. In the remainder of this paper, 0.05 is arbitrarily set as a threshold where the corresponding $\|d\|$ is regarded as the analogous radius to visually illustrate the situations of combat fields.

Moreover, it is assumed that the UCAV maintains a constant speed and thus the fuel consumption corresponds with the total length of flight path. Let $\|ST\|$ denote the length of segment ST, the total cost of the flight path is defined in (3), where $\int d_j$ refers to the length of the flight path and apparently. γ is a weighting factor ranging from 0 to 1, and in the remainder of this paper, it is arbitrarily set that $\gamma = 0.5$.

$$C_{\text{total cost}} = [\exp(-\frac{\sum_{i=1}^{n} \|d_i\|}{\delta})] \cdot \gamma + \frac{\int d_j}{\|ST\|} \cdot (1-\gamma) \tag{3}$$

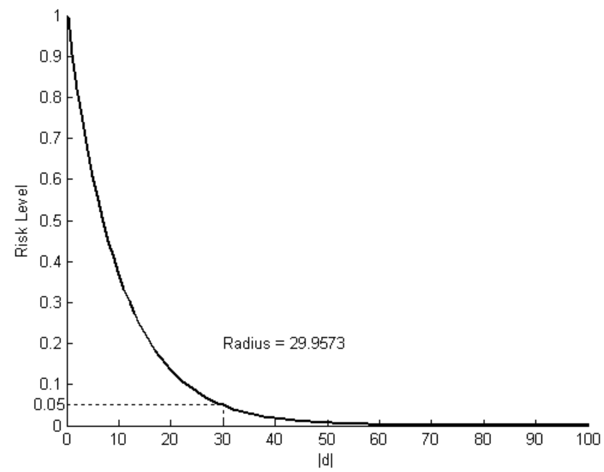

Fig. 2. Visualization of the proposed cost function in probability density model with δ= -10 . Note that as the distance $|d|$ increases, the probability of suffering a damage risk gradually decreases but it would never be zero in this novel model.

By using this model, the path planning task is transformed into a numerical problem for its optimal solution.

III. Principle of Artificial Bee Colony Algorithm

The algorithm of ABC is an iterative optimization process [18], in which the bee colony consists of three groups: employed bees, onlooker bees, and scout bees. The position of i-th food source X_i represents a possible solution to the optimization problem and the profitability P_i corresponds to the quality (fitness) of the solution. The number of employed bees SN is equal to that of onlookers, and that of food sources. Any employed bee cannot improve its position further in some certain cycles will be replaced with a scout bee. At first, SN employed bees start to explore randomly in the environment. I.e., SN uniformly distributed food source positions are generated for these employed bees. The process to initially generate j-th vector for i-th solution X_i^j is represented by the equation below,

$$X_i^j = X_{\min}^j + rand(0,1) \cdot (X_{\max}^j - X_{\min}^j), \quad j = 1,2,\ldots,D,\ i = 1,2,\ldots,SN \tag{4}$$

where $X_{\min}^j$ and $X_{\min}^j$ denote defined boundaries of this vector.

In each iteration, each employed bee executes the crossover process to share information with a neighbor and refresh their positions by (5),

$$X_i^j \leftarrow X_i^j + rand(0,1)\cdot(X_k^j - X_i^j) \tag{5}$$

where i-th employed bee implements the crossover process with k-th one in j-th vector. It should be noted that only one vector is changed for each employed bee in each iteration and the selection of *j* or *k* obeys uniform distribution within their limits respectively. If *k* equals *i*, *k* is regenerated until it does not. Greedy strategy is implemented to choose between current and previous X_i^j.

Afterwards, employed bees share their nectar amount information with the onlooker bees in the nearby hive. Each onlooker randomly choose to exploit or not around the corresponding employed bee's position with the probability of P_i, which is calculated by the following equation,

$$P_i = \frac{\mathrm{obj}(X_i)}{\sum_{i=1}^{SN}\mathrm{obj}(X_i)} \tag{6}$$

where $\mathrm{obj}(X_i)$ refers to the value obtained by substituting X_i into the objective function. It is obvious that higher $\mathrm{obj}(X_i)$ enjoys higher probability of being selected by onlooker bees. Then, qualified P_i guide corresponding onlooker bees to exploit. Such approach is known as the famous roulette selection strategy.

For the working onlooker bees, new positions are generated around their selected food sources also by (5). Similarly, the greedy strategy is applied between the current X_i and previous one for each onlooker bee.

During the searching process, each employed bee together with the corresponding onlooker bee memories the iterations of no avail by $trial(i)$, $i = 1,2,...,SN$. If a better position is obtained by *i*-th employed bee or *i*-th onlooker bee, $trial(i)$ is set zero. Before the whole iteration process, a prior parameter $limit$ is arbitrarily set as a threshold. If $trial(i) \geq limit$, current *i*-th position is abandoned a scout bee takes its place with randomly initialized position by (4). It is noted that no more than one scout bee is allowed to emerge in each iteration.

The pseudo-code of ABC for constrained optimization problems is given below.

Initialize solution population by (4)
repeat
  while $iter \leq MCN$, **do**
    generate positions for employed bees by (5)
    evaluate and greedily select employed bees
    if position is improved
      $trial(i) \leftarrow 0$
    end if
    calculate P_i by (6)
    if $P_i > rand(0,1)$, **do**
      generate positions for onlooker bees by (5)
      evaluate and greedily select onlookers
      if position is improved
        $trial(i) \leftarrow 0$
      else
        $trial(i) \leftarrow trial(i)+1$
      end if
    end if
    if $trial(i) \geq limit$, **do**
      initialize the position by (4)
    end if
    memorize current best solution
    $iter \leftarrow iter+1$
  end while
end repeat
output global optimum

IV. EXPERIMENTAL RESULTS AND DISCUSSION

To evaluate their qualities, PSO, ABC, and C-ABC are applied to various cases. All simulations are implemented in MATLAB R2010a and executed on an Intel Core 2 Due CPU with 2 GB RAM running at 2.53 GHz under Windows XP. Let $limit = 10\% \times MCN$ for ABC, both acceleration coefficients equal 1.4962 and inertia weight equals 0.7298 for PSO implemented in this work. Logistic equation is chosen for chaos in C-ABC. The swarm population is arbitrarily set to be 40 for all experiments and each type of simulation repeats 30 times with randomly initialized conditions.

To investigate the initial efficiency of different algorithms, experiments are conducted to test on convergence rate by 1.00 second. Fig. 3 presents the convergence curves obtained by ABC, PSO, as well as C-ABC under the condition of *D*=20. It is implied that the C-ABC leads to more iterations executed within 1.00 second in comparison with ABC, which results from the fact that the command of *rand* in MATLAB usually takes far more time than chaos strategy to generate a random number. Although the convergence rate of C-ABC resembles that of ABC in this case, it is noted that chaos strategy requires less time dealing with real-time tasks.

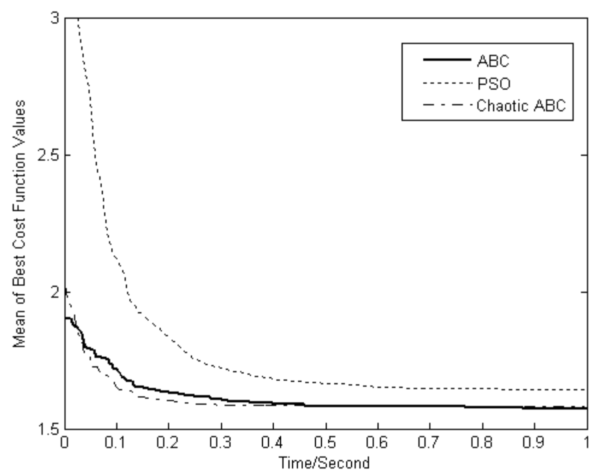

Fig. 3. Comparison of 1-second-converging performances for optimal solutions (*D*=20). Note that both ABC and C-ABC are superior to PSO in this case.

Fig. 4 demonstrates the corresponding paths in a combat filed, where the solid curve refers to the optimal path obtained by ABC, and the dashed curve by PSO. In this figure, it is verified that even for the problem with relatively lower dimension, ABC is capable to obtain smoother and more advantageous path than PSO.

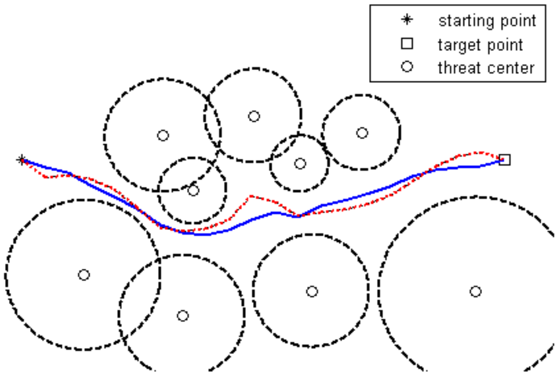

Fig. 4. Comparison of 1-second-paths obtained by ABC and PSO in the combat field (D=20). In this figure, the solid curve refers to the optimal path obtained by ABC, and the dashed one by PSO. It is confirmed that ABC helps to obtain a smoother path in comparison with PSO. But the curve optimized by C-ABC is not drawn due to its resemblance to the curve optimized by ABC.

To further reveal the convergence rate, simulations of 100 iterations are conducted to compare ABC and PSO. Specifically, the simulations are repeated 100 times each, and the standard division is calculated as shown by the doubled error bars plotted in Fig. 5 and 6.

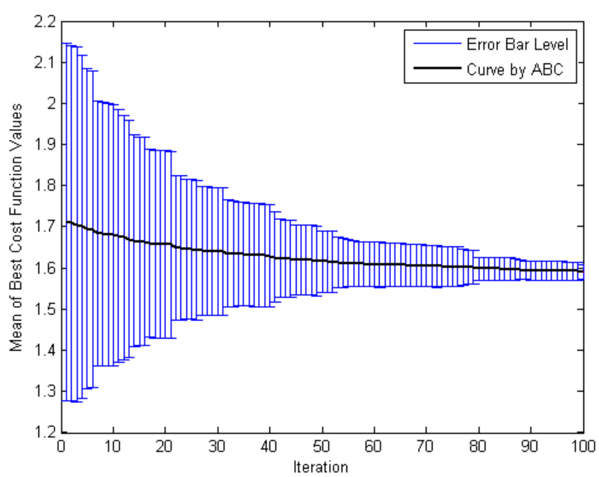

Fig. 5. Convergence curve and the corresponding variance information for iteration up to 100 (D=20). The bold curve denotes convergence curve by ABC and thin error bars onto the curve denotes doubled standard deviation in statistics to reveal the robustness of convergence.

In Fig. 5, the initial dynamics gradually dies out in a more stable way using ABC, which indicates a robust convergence rate, in comparison with those using PSO in Fig. 6. Therefore, it is safely concluded that, for UCAV path planning problem, ABC is superior to PSO in efficiency as well as convergence rate.

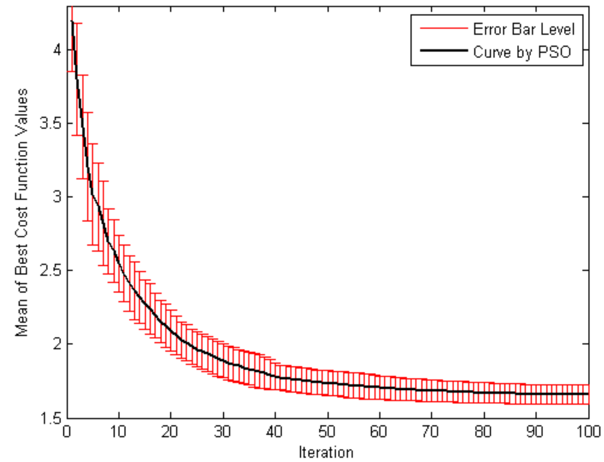

Fig. 6. Convergence curve and the corresponding variance information for iteration up to 100 (D=20). The bold curve denotes convergence curve by PSO and thin error bars onto the curve denotes doubled standard deviation in statistics to reveal the robustness of convergence.

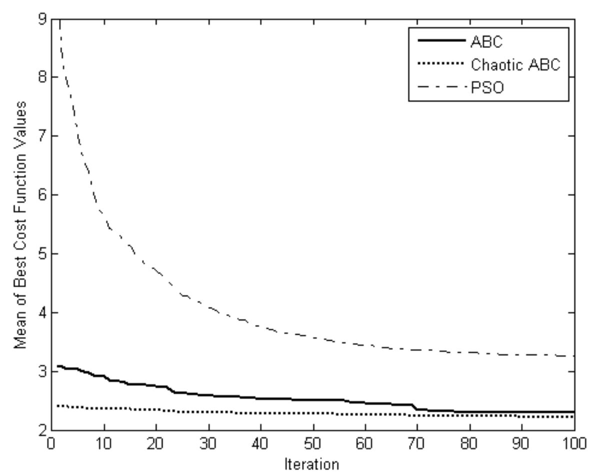

Fig. 7. Comparison of 100-iteration-converging performances for optimal solution (D=40). It is noted that as dimension increases, PSO obviously lags in convergence rate.

Additionally, the capacities of ABC, C-ABC, and PSO are investigated to settle cases with higher dimensions. As D grows to 40, in general, convergence performance deteriorates as in Fig. 7. Note that C-ABC performs slightly better than ABC, both of which show great superiority over PSO. Also, C-ABC converges faster here, which, results from the efficient initial swarm produced by chaos strategy. That is, C-ABC sacrifices its original dynamics, subsequent effectiveness and robustness so as to get a more efficient initial swarm, which, however, may not be an advisable bargain. Fig. 8 demonstrates the corresponding paths, where the dotted curve refers to solution by PSO and the other by ABC. Several similar comparisons are made as shown in Fig. 9–12. It is noted that paths obtained by using ABC are always smoother than ones by PSO, as shown specifically in Fig. 9 and 11. Regarded with the case illustrated in Fig. 10, the optimization process by using PSO is trapped in a local optimum.

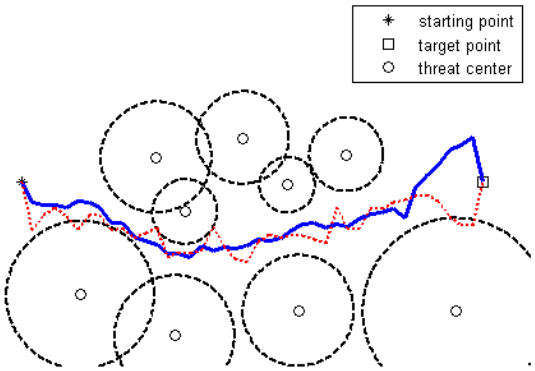

Fig. 8. Comparison of 100-ieration-paths obtained by ABC and PSO in the combat field (D=40). In this figure, the solid curve refers to the optimal path obtained by ABC, and the dashed one by PSO.

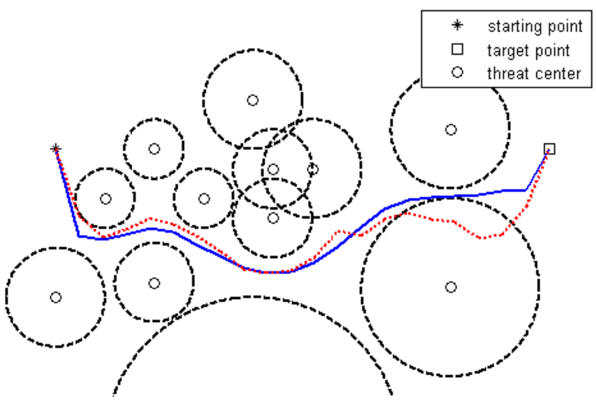

Fig. 9. Comparison of 1000-ieration-paths obtained by ABC and PSO in the combat field (D=20). In this figure, the solid curve refers to the optimal path obtained by ABC, and the dashed one by PSO. It is noted that path optimized by ABC is smoother than that by PSO.

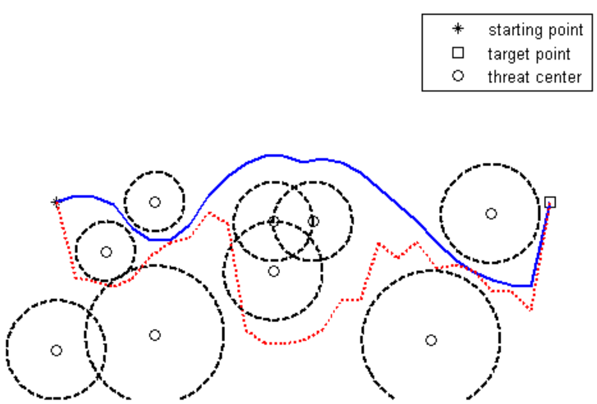

Fig. 10. Comparison of 1000-ieration-paths obtained by ABC and PSO in the combat field (D=25). In this figure, the solid curve refers to the optimal path obtained by ABC, and the dashed one by PSO. It is obvious that the optimization process by using PSO is trapped in a local optimum.

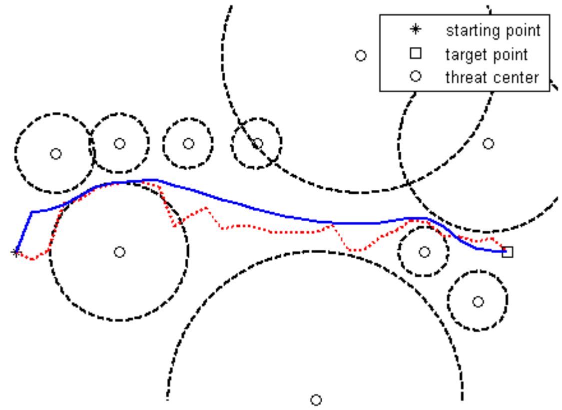

Fig. 11. Comparison of 1000-ieration-paths obtained by ABC and PSO in the combat field (D=30). In this figure, the solid curve refers to the optimal path obtained by ABC, and the dashed one by PSO. It is noted that path optimized by ABC is smoother than that by PSO.

It is confirmed that probability density model effectively reflects the true situations in combat fields. Regarded with the cases in Fig. 12, it is inevitable that the UCAV will pass through several threat circles before it arrives at the target point. By proposing the probability density model, we aim to minimize the risk of selecting a path, but not to avoid every threat circle in an inflexible way.

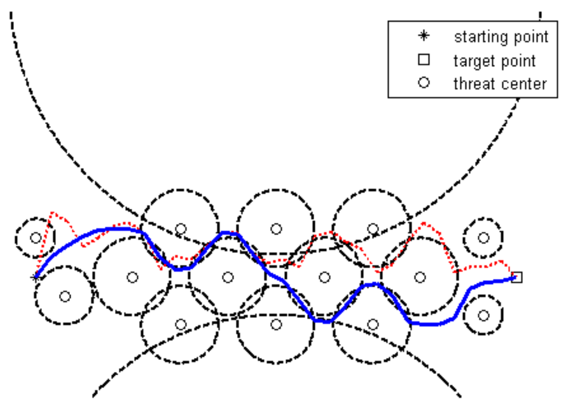

Fig. 12. Comparison of 1000-ieration-paths obtained by ABC and PSO in the combat field (D=30). In this figure, the solid curve refers to the optimal path obtained by ABC, and the dashed one by PSO. It is noted that the proposed probability density model is superior to the tradition circle threat model.

V. CONCLUSION

This paper intends to authenticate the actual competence of ABC to deal with UCAV path planning problem in comparison with some other swarm intelligence algorithms. It is noted that the study only concentrates on several cases in 2D surface and future effort is still needed to verify the capacity of ABC for cases in 3D space. In addition, the selection of weighting factor in the cost function remains to be an outstanding question in our model. By implementing on multi-core CPUs, it is possible to obtain feasible real-time paths for UCAV using ABC. In general, experimental results in the current work confirm that, with statistical significance, paths obtained by ABC are superior to those by other intelligence algorithms regarding convergence rate, efficiency, and robustness.

ACKNOWLEDGMENT

Bai Li thanks Professor H. B. Duan for his initial inspiration to carry on the research.

REFERENCES

[1] Wang, Gaige, et al. "A Hybrid Metaheuristic DE/CS Algorithm for UCAV Three-Dimension Path Planning." *The Scientific World Journal* 2012 (2012).

[2] Han-Qiao, Huang, et al. "Research on online and real-time path planning of UCAV by tracking and controlling in multi-restriction." *Control Conference (CCC), 2012 31st Chinese*. IEEE, 2012.

[3] Ma, Qianzhi, and Xiujuan Lei. "Application of artificial fish school algorithm in UCAV path planning." *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on*. IEEE, 2010.

[4] Macharet, Douglas, Armando Neto, and Mario Campos. "Feasible UAV path planning using genetic algorithms and Bézier curves." *Advances in Artificial Intelligence–SBIA 2010* (2011): 223-232.

[5] Holland, John H. "Adaptation in natural and artificial systems, University of Michigan press." *Ann Arbor, MI* 1.97 (1975): 5.

[6] Storn, Rainer, and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." *Journal of global optimization* 11.4 (1997): 341-359.

[7] Eberhart, Russell, and James Kennedy. "A new optimizer using particle swarm theory." *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*. IEEE, 1995.

[8] Karaboga, Dervis. "An idea based on honey bee swarm for numerical optimization." *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes* (2005).

[9] Xu, Chunfang, Haibin Duan, and Fang Liu. "Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle (UCAV) path planning."*Aerospace Science and Technology* 14.8 (2010): 535-541.

[10] Ye, Wen, Deng-Wu Ma, and Hong-Da Fan. "Algorithm for low altitude penetration aircraft path planning with improved ant colony algorithm."*Chinese Journal of Aeronautics* 18.4 (2005): 304-309.

[11] Zheng, Changwen, et al. "Evolutionary route planner for unmanned air vehicles." *Robotics, IEEE Transactions on* 21.4 (2005): 609-620.

[12] Macharet, Douglas, Armando Neto, and Mario Campos. "Feasible UAV path planning using genetic algorithms and Bézier curves." *Advances in Artificial Intelligence–SBIA 2010* (2011): 223-232.

[13] Cheng, Ze, Ying Sun, and Yanli Liu. "Path planning based on immune genetic algorithm for UAV." *Electric Information and Control Engineering (ICEICE), 2011 International Conference on*. IEEE, 2011.

[14] Bao, Yong, Xiaowei Fu, and Xiaoguang Gao. "Path planning for reconnaissance UAV based on particle swarm optimization." *Computational Intelligence and Natural Computing Proceedings (CINC), 2010 Second International Conference on*. Vol. 2. IEEE, 2010.

[15] Fu, Yangguang, Mingyue Ding, and Chengping Zhou. "Phase Angle-Encoded and Quantum-Behaved Particle Swarm Optimization Applied to Three-Dimensional Route Planning for UAV." *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 42.2 (2012): 511-526.

[16] Pierre, Djamalladine Mahamat, Nordin Zakaria, and Anindya Jyoti Pal. "Master-Slave Parallel Vector-Evaluated Genetic Algorithm for Unmanned Aerial Vehicle's path planning." *Hybrid Intelligent Systems (HIS), 2011 11th International Conference on*. IEEE, 2011.

[17] Roberge, Vincent, Mohammed Tarbouchi, and Gilles Labonté. "Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real Time UAV Path Planning." *Industrial Informatics, IEEE Transactions on* 9.2 (2013): 132-141.

[18] Li, Bai, and Li Ya. "BE-ABC: Hybrid artificial bee colony algorithm with balancing evolution strategy." *Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference on*. IEEE, 2012.