Predator-Prey Pigeon-Inspired Optimization for UAV Three-Dimensional Path Planning

Bo Zhang[1] and Haibin Duan[1,2,*]

[1] Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, P. R. China
[2] Provincial Key Laboratory for Information Processing Technology, Soochow University, Suzhou, 215006, P. R. China
zhangbo0216@163.com, hbduan@buaa.edu.cn

Abstract. Pigeon-inspired optimization (PIO) is a new bio-inspired optimization algorithm. This algorithm searches for global optimum through two models: map and compass operator model is presented based on magnetic field and sun, while landmark operator model is designed based on landmarks. In this paper, a novel Predator-prey pigeon-inspired optimization (PPPIO) is proposed to solve the three-dimensional path planning problem of unmanned aerial vehicles (UAVs), which is a key aspect of UAV autonomy. To enhance the global convergence of the PIO algorithm, the concept of predator-prey is adopted to improve global best properties and enhance the convergence speed. The comparative simulation results show that our proposed PPPIO algorithm is more efficient than the basic PIO and particle swarm optimization (PSO) in solving UAV three-dimensional path planning problems.

Keywords: pigeon-inspired optimization (PIO), unmanned aerial vehicle (UAV), path planning, predator-prey.

1 Introduction

Three-dimensional path planner is an essential element of the unmanned aerial vehicle (UAV) autonomous control module [1]. It allows the UAV to compute the best path from a start point to an end point autonomously [2, 3]. Whereas commercial airlines fly constant prescribed trajectories, UAVs in operational areas have to travel constantly changing trajectories that depend on the particular terrain and conditions prevailing at the time of their flight.

Pigeon-inspired optimization (PIO), which is a new swarm intelligence optimizer based on the movement of pigeons, was firstly invented by Duan in 2014 [4]. Homing pigeons can easily find their homes by using three homing tools: magnetic field, sun and landmarks. In the optimization, map and compass model is presented based on magnetic field and sun, while landmark operator model is presented based on landmarks.

In this paper, we propose a predator-prey pigeon-inspired optimization (PPPIO) method, integrating the concept of predator-prey into PIO in order to improve its

* Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 96–105, 2014.

capability of finding satisfactory solutions and increasing the diversity of the population. We also solve the UAV three-dimensional path planning problem by PPPIO. Simulation results and comparisons verified the feasibility and effectiveness of our proposed algorithm.

The rest of the paper is organized as follows: Section 2 provides the representation and the cost function we developed to evaluate the quality of candidate trajectories. Section 3 describes the principle of basic PIO algorithm. Section 4 shows the implementation procedure of our proposed predator-prey PIO algorithm. Finally, we compare the quality of the trajectories produced by the PIO, particle swarm optimization (PSO) and the PPPIO in Section 5.

2 Problem Formulation

The first step of three-dimensional path planning is to discretize the world space into a representation that will be meaningful to the path planning algorithm. In this work, we use a formula to indicate the terrain environment. The mathematical function is of the form [5]:

$$z(x, y) = \sin(x/5+1) + \sin(y/5) + \cos(a \bullet \sqrt{x^2+y^2}) + \sin(b \bullet \sqrt{x^2+y^2}) \tag{1}$$

where z indicate the altitude of a certain point, and a, b are constants experimentally defined. Our representation of cylindrical danger zones (or no-fly zones) to be in a separate matrix where each row represents the coordinates (x_i, y_i) and the radius r_i of the ith cylinder as shown in Eq. (2). Complex no-fly zone can be built by partially juxtaposing multiple cylinders

$$\text{danger zones} = \begin{pmatrix} x_1 & y_1 & r_1 \\ x_2 & y_2 & r_2 \\ \cdots & \cdots & \cdots \\ x_n & y_n & r_n \end{pmatrix} \tag{2}$$

The three-dimensional trajectories generated by the algorithm are composed of line segments and (x_i, y_i, z_i) represents the coordinates of the ith way point. The trajectories are flown at constant speed.

In the situation of UAV path planning, the optimal path is complex and includes many different characteristics. To take into account these desired characteristics, a cost function is used and the path planning algorithm becomes a for a path that will minimize the cost function. We define our cost function as follows [6]:

$$F_{\text{cost}} = C_{\text{length}} + C_{\text{altitude}} + C_{\text{danger zones}} + C_{\text{power}} + C_{\text{collision}} + C_{\text{fuel}} \tag{3}$$

In the cost function, the term associated with the length of a path is defined as follows:

$$C_{\text{length}} = 1 - \left(\frac{L_{\text{p1p2}}}{L_{\text{traj}}}\right) \tag{4}$$

$$C_{\text{length}} \in [0,1] \tag{5}$$

where L_{p1p2} is the length of the straight line connecting the starting point $P1$ and the end point $P2$ and L_{traj} is the actual length of the trajectory.

The term associated with the altitude of the path is defined as follows:

$$C_{\mathrm{altitude}} = \frac{A_{\mathrm{traj}} - Z_{\mathrm{min}}}{Z_{\mathrm{max}} - Z_{\mathrm{min}}} \tag{6}$$

$$C_{\mathrm{altitude}} \in [0,1] \tag{7}$$

where Z_{max} is the upper limit of the elevation in our search space, Z_{min} is the lower limit and A_{traj} is the average altitude of the actual trajectory. Z_{max} and Z_{min} are respectively set to be slightly above the highest and lowest point of the terrain.

The term associated with the violation of the danger zones is defined as follows:

$$C_{\mathrm{danger\ zones}} = \frac{L_{\mathrm{inside\ d.z.}}}{\sum_{i=1}^{n} d_i} \tag{8}$$

$$C_{\mathrm{danger\ zones}} \in [0,1] \tag{9}$$

where n is the total number of danger zones, $L_{\mathrm{inside\ d.z.}}$ is the total length of the subsections of the trajectory which go through danger zones and d_i is the diameter of the danger zone i.

The term associated with a required power higher than the available power of the UAV is defined as follows:

$$C_{\mathrm{power}} = \begin{cases} 0, & L_{\mathrm{not\ feasible}} = 0 \\ P + \left(\dfrac{L_{\mathrm{not\ feasible}}}{L_{\mathrm{traj}}} \right), & L_{\mathrm{not\ feasible}} > 0 \end{cases} \tag{10}$$

$$C_{\mathrm{power}} \in 0 \bigcup [P, P+1] \tag{11}$$

where $L_{\mathrm{not\ feasible}}$ is the sum of the lengths of the line segments forming the trajectory which require more power than the available power of the UAV, L_{traj} is the total length of the trajectory and P is the penalty constant. This constant must be higher than the cost of the worst feasible trajectory which would have, based on our cost function, a cost of 3. By adding this penalty P, we separate nonfeasible solutions from the feasible ones.

The term associated with ground collisions is defined as follows:

$$C_{\mathrm{collision}} = \begin{cases} 0, & L_{\mathrm{under\ terrain}} = 0 \\ P + \left(\dfrac{L_{\mathrm{under\ terrain}}}{L_{\mathrm{traj}}} \right), & L_{\mathrm{under\ terrain}} > 0 \end{cases} \tag{12}$$

$$C_{collision} \in 0 \bigcup [P, P+1] \tag{13}$$

where $L_{\text{under terrain}}$ is the total length of the subsections of the trajectory which travels below the ground level and L_{traj} is the total length of the trajectory.

The term associated with an insufficient quantity of fuel available is defined as follows:

$$C_{\text{fuel}} = \begin{cases} 0, & \mathrm{F}_{\text{traj}} \le F_{\text{init}} \\ P+1-\left(\dfrac{F_{\text{P1P2}}}{F_{\text{traj}}}\right), & \mathrm{F}_{\text{traj}} > F_{\text{init}} \end{cases} \tag{14}$$

$$C_{\text{fuel}} \in 0 \bigcup [P, P+1] \tag{15}$$

where F_{P1P2} is the quantity of fuel required to fly the imaginary straight segment connection the starting point $P1$ to the end point $P2$, F_{traj} is the actual amount of fuel needed to fly the trajectory, F_{init} is the initial quantity of fuel on board the UAV.

The search engine will be adopted to find a solution, which can minimize the cost function during the optimization phase of our path planner algorithm. This can also be explained as to find a trajectory that best satisfies all the qualities represented by this cost function. Our cost function demonstrates a specific scenario where the optimal path minimizes the distance travelled, the average altitude (to increase the stealthiness of the UAV) and avoids danger zones, while respecting the UAV performance characteristics. This cost function is highly complex and demonstrates the power of our path planning algorithm. However, this cost function could easily be modified and applied to a different scenario.

3 Principle of Basic PIO

PIO is a novel swam intelligence optimizer for solving global optimization problems. It is based on natural pigeon behavior. Studies show that the species seem to have a system in which signals from magnetite particles are carried from the nose to the brain by the trigeminal nerve [4, 7]. Evidence that the sun is also involved in pigeon navigation has been interpreted, either partly or entirely, in terms of the pigeon's ability to distinguish differences in altitude between the Sun at the home base and at the point of release [8]. Recent researches on pigeons' behaviors also show that the pigeon can follow some landmarks, such as main roads, railways and rivers rather than head for their destination directly. The migration of pigeons is summarized as two mathematical models. One is map and compass operator, and the other is landmark operator.

3.1 Map and Compass Operator

In PIO model, virtual pigeons are used. In the map and compass operator, the rules are defined with the position X_i and the velocity V_i of pigeon i, and the positions and velocities in a *D*-dimension search space are updated in each iteration. The new position X_i and velocity V_i of pigeon i at the t-th iteration can be calculated with the follows [3]:

$$V_i(t) = V_i(t-1) \cdot e^{-Rt} + rand \cdot (X_g - X_i(t-1)) \tag{16}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{17}$$

where R is the map and compass factor, *rand* is a random number, and X_g is the current global best position, and which can be obtained by comparing all the positions among all the pigeons.

3.2 Landmark Operator

In the landmark operator, half of the number of pigeons is decreased by N_p in every generation. However, the pigeons are still far from the destination, and they are unfamiliar the landmarks. Let $X_c(t)$ be the center of some pigeons' position at the t-th iteration, and suppose every pigeon can fly straight to the destination. The position updating rule for pigeon i at t-th iteration can be given by:

$$N_P(t) = \frac{N_P(t-1)}{2} \tag{18}$$

$$X_c(t) = \frac{\sum X_i(t) \cdot fitness\left(X_i(t)\right)}{N_P \sum fitness\left(X_i(t)\right)} \tag{19}$$

$$X_i(t) = X_i(t-1) + rand \cdot (X_c(t) - X_i(t-1)) \tag{20}$$

where *fitness* is the quality of the pigeon individual. For the minimum optimization problems, we can choose $fitness\left(X_i(t)\right) = \frac{1}{f\left(X_i(t)\right) + \varepsilon}$ for maximum optimization problems, we can choose $fitness\left(X_i(t)\right) = f\left(X_i(t)\right)$.

4 PPPIO for Three-Dimensional Path Planning

4.1 Predator-Prey Concept

Predatory behavior is one of the most common phenomena in nature, and many optimization algorithms are inspired by the predator-prey strategy from ecology [9]. In nature, predators hunt prey to guarantee their own survival, while the preys need to be able to run away from predators. On the other hand, predators help to control the prey population while creating pressure in the prey population. In this model, an individual in predator population or prey population represents a solution, each prey in the population can expand or get killed by predators based on its fitness value, and a predator always tries to kill preys with least fitness in its neighborhood, which represents removing bad solutions in the population. In this paper, the concept of predator-prey is used to increase the diversity of the population, and the predators are modeled based on the worst solutions which are demonstrated as follows:

$$P_{\text{predator}} = P_{\text{worst}} + \rho(1 - t / t_{\max}) \qquad (\mathbf{21})$$

where P_{predator} is the predator (a possible solution), P_{worst} is the worst solution in the population, t is the current iteration, while $t_{\max}$ is the maximum number of iterations and ρ is the hunting rate. To model the interactions between predator and prey, the solutions to maintain a distance of the prey from the predator is showed as follows:

$$\begin{cases} P_{\text{k+1}} = P_{\text{k}} + \rho e^{-|d|}, & \text{d>0} \\ P_{\text{k+1}} = P_{\text{k}} - \rho e^{-|d|}, & \text{d<0} \end{cases} \qquad (\mathbf{22})$$

where d is the distance between the solution and the predator, and k is the current iteration.

4.2 Parallelization of the Map and Compass Operations and the Landmark Operations

In the basic model of PIO algorithm, the landmark operation is used after several iterations of map and compass operation. For example, when the number of generations N_c is larger than the maximum number of generations of the map and compass operation $N_{c\max 1}$. The map and compass operator will stop and it the landmark operation will be start. During my experiment, we found it's easy to fall into a local best solution before the number of generations got to $N_{c\max 1}$. Furthermore, half of the number of pigeons is decreased by N_p in every generation on the landmark operator. The population of pigeons is decreased too rapidly according to formula (18), which would reach to zero after a small amount of iterations. The landmark operator would make only a small impact on the pigeons' position by this way. So we make a small modification on the basic PIO algorithm. The map and compass operation and the compass operation are used parallelly at each iteration. A parameter ω is used to define the impaction of the landmark increase with a smoothly path. And a constant parameter c is used to define the number of pigeons that are in the landmark operator. Our new formula of landmark operator is as follows:

$$N_P(t) = c \cdot N_{P\max} \qquad \text{c} \in (0,1) \qquad (\mathbf{23})$$

$$X_c(t) = \frac{\sum X_i(t) \cdot fitness\left(X_i(t)\right)}{N_P \sum fitness\left(X_i(t)\right)} \qquad (\mathbf{24})$$

$$\omega = s + (1 - \mathrm{s}) \cdot \mathrm{t} / \mathrm{N}_{c\max} \qquad \mathrm{s} \in (0,1) \tag{25}$$

$$X_i(t) = X_i(t-1) + \omega \cdot rand \cdot (X_c(t) - X_i(t-1)) \tag{26}$$

where s is a constant experimentally defined.

4.3 Proposed Predator-Prey PIO (PPPIO) Based Path Planner

In order to overcome the disadvantages of the classical PIO algorithm, such as the tendency to converge to local best solutions, PPPIO, which integrates PIO with the concept of predator-prey, was proposed in our work. After the mutation of each generation, the predator-prey behavior is been conducted in order to choose better solutions into next generation. In this way, our proposed algorithm takes the advantage of the predator-prey concept to make the individuals of sub generations distributed ergodically in the defined space and it can avoid from the premature of the individuals, as well as to increase the speed of finding the optimal solution.

The implementation procedure of our proposed PIO approach to UAV 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.

Step 2: Initialize the PIO parameters, such as solution space dimension *D*, the population size N_p, map and compass factor R, the number of iteration N_c.

Step 3: Set each pigeon with a randomized velocity and path. Compare the fitness of each pigeons, and find the current best path.

Step 4: Operate map and compass operator. Firstly, we update the velocity and path of every pigeon by using Eqs. (16) and (17).

Step 5: Rank all pigeons according their fitness values. Some of pigeons whose fitness are low will follow those pigeons with high fitness according to Eq. (23). We then find the center of all pigeons according to Eq. (24), and this center is the desirable destination. All pigeons will fly to the destination by adjusting their flying direction according to Eq. (26). Next, store the best solution parameters and the best cost value.

Step 6: Model the predators based on the worst solution as Eq. (15) demonstrates. Then, use Eq. (16) to provide the other solutions to maintain a distance between the predator and the prey.

Step 7: If $Nc > N_{c\max}$, stop the iteration, and output the results. If not, go to step 6.

5 Comparative Experimental Results

In order to evaluate the performance of our proposed PPPIO algorithm in this work, series of experiments are conducted in Matlab2012a programing environment. Coordinates of a starting point are set as (10, 16, 0), and the target point as (55, 100, 0). The initial parameters of PIO algorithm were set as: NP =150. The comparative results of PPPIO with PIO and PSO are showed as follows:

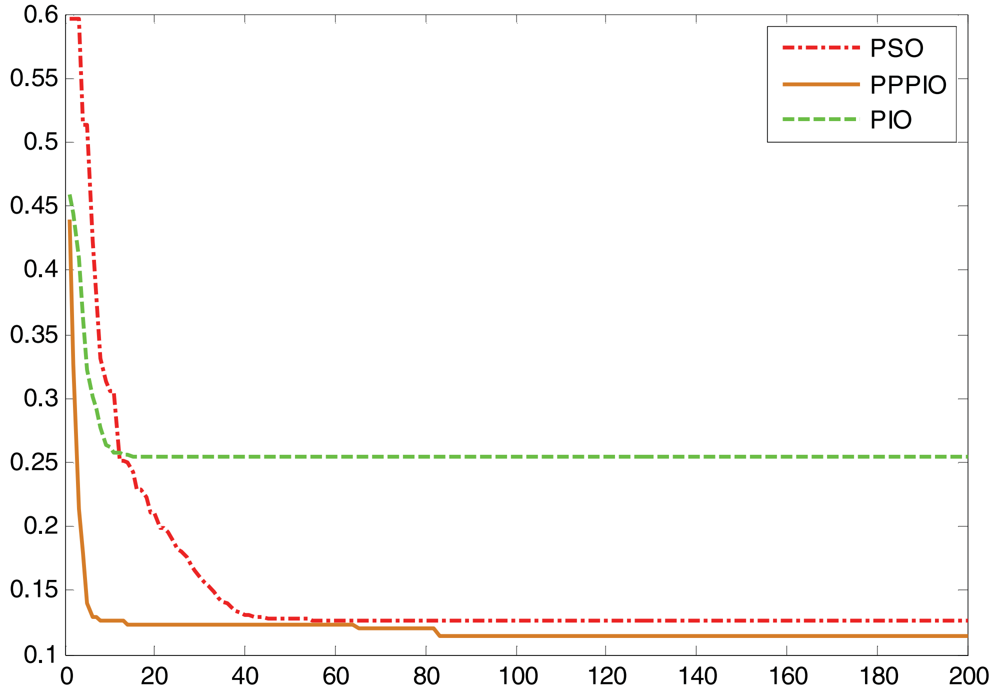

Fig. 1. Comparative evolutionary curves of PPPIO, PIO and PSO

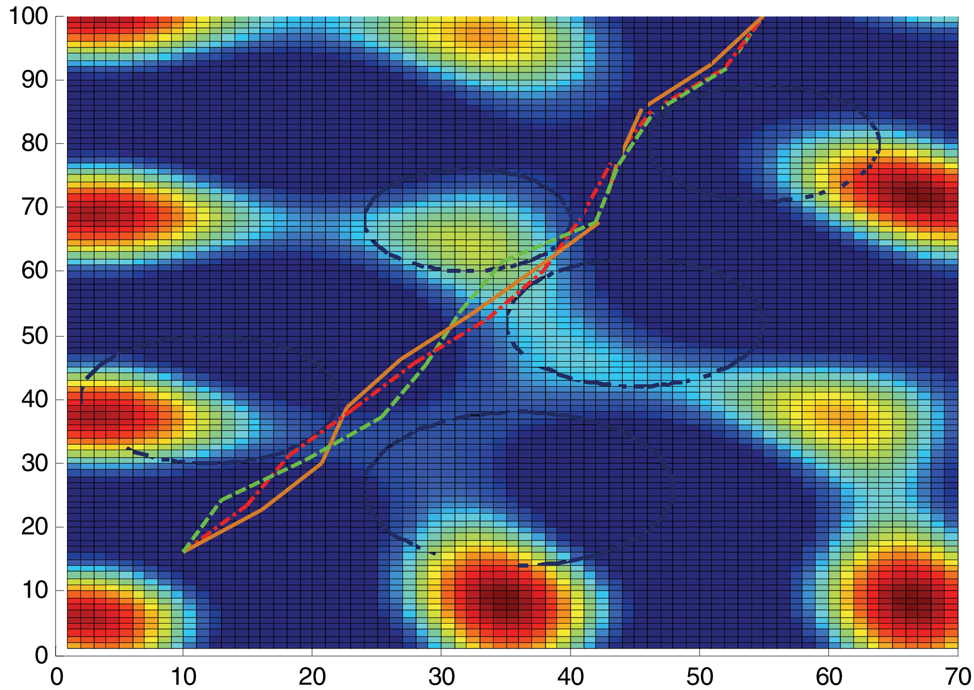

Fig. 2. Comparative path planning results of PPPIO, PIO and PSO

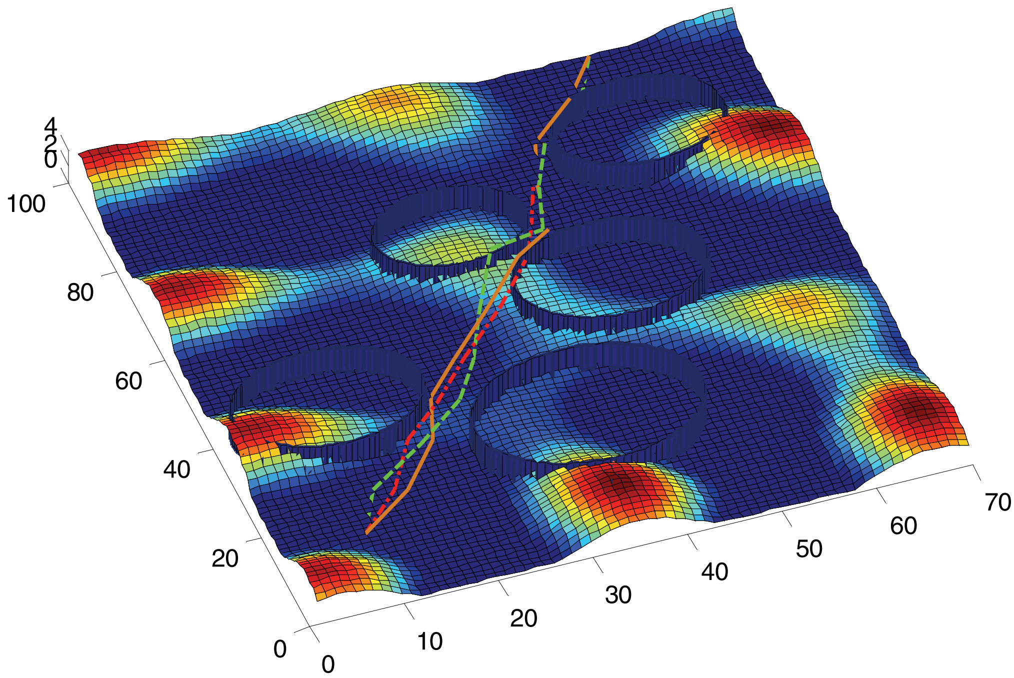

Fig. 3. Comparative path planning results of PPPIO, PIO and PSO on 3D version

6 Conclusions

This paper proposed a novel PPPIO algorithm for solving the UAV three-dimensional path planning problem in complex environments. The concept of predator-prey is adopted to improve the performance of the basic PIO algorithm. Series of comparative simulation results were given to show that our proposed PPPIO algorithm is more efficient than basic PIO and PSO in solving UAV three-dimensional path planning problems.

Acknowledgements. This work was partially supported by National Key Basic Research Program of China(973 Project) under grant #2014CB046401, Natural Science Foundation of China (NSFC) under grant # 61333004 and #61273054, National Magnetic Confinement Fusion Research Program of China under grant # 2012GB102006, and Aeronautical Foundation of China under grant #20135851042.

References

1. Chen, H., Wang, X.M., Li, Y.: A Survey of Autonomous Control for UAV. In: International Conference on Artificial Intelligence and Computational Intelligence, vol. 2, pp. 267–271 (2009)
2. Duan, H.B., Li, P.: Bio-inspired Computation in Unmanned Aerial Vehicles. Springer, Heidelberg (2014)

3. Duan, H.B., Luo, Q.N., Ma, G.J., Shi, Y.H.: Hybrid Particle Swarm Optimization and Genetic Algorithm for Multi-UAVs Formation Reconfiguration. IEEE Computational Intelligence Magazine 8(3), 16–27 (2013)
4. Duan, H.B., Qiao, P.X.: Pigeon-Inspired Optimization: A New Swarm Intelligence Optimizer for Air Robot Path Planning. International Journal of Intelligent Computing and Cybernetics 7(1), 24–37 (2014)
5. Ioannis, K.N., Athina, N.B.: Coordinated UAV Path Planning Using Differential Evolution. In: IEEE International Symposium on, Mediterrean Conference on Control and Automation, vol. 70, pp. 77–111. Springer, Heidelberg (2005)
6. Vincent, R., Mohammed, T., Gilles, L.: Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. IEEE Transactions on Industrial Informatics 9(1), 132–141 (2013)
7. Mora, C.V., Davison, M., Wild, J.M., Michael, M.W.: Magnetoreception and Its Trigeminal Mediation in the Homing Pigeon. Nature 432, 508–511 (2004)
8. Whiten, A.: Operant Study of Sun Altitude and Pigeon Navigation. Nature 237, 405–406 (1972)
9. Zhu, W.R., Duan, H.B.: Chaotic Predator-Prey Biogeography-Based Optimization Approach for UCAV Path Planning. Aerospace Science and Technology 32(1), 153–161 (2014)