algorithms

Article

Adaptive Operator Quantum-Behaved Pigeon-Inspired Optimization Algorithm with Application to UAV Path Planning

Chunhe Hu †**, Yu Xia** † **and Junguo Zhang** *

School of Technology, Beijing Forestry University, Beijing 100083, China; huchunhe@bjfu.edu.cn (C.H.); xyodfish0711@bjfu.edu.cn (Y.X.)

* Correspondence: zhangjunguo@bjfu.edu.cn; Tel.: +86-10-62336398

† These authors contributed equally to this work.

Received: 23 November 2018; Accepted: 20 December 2018; Published: 21 December 2018

Abstract: Path planning of unmanned aerial vehicles (UAVs) in threatening and adversarial areas is a constrained nonlinear optimal problem which takes a great amount of static and dynamic constraints into account. Quantum-behaved pigeon-inspired optimization (QPIO) has been widely applied to such nonlinear problems. However, conventional QPIO is suffering low global convergence speed and local optimum. In order to solve the above problems, an improved QPIO algorithm, adaptive operator QPIO, is proposed in this paper. Firstly, a new initialization process based on logistic mapping method is introduced to generate the initial population of the pigeon-swarm. After that, to improve the performance of the map and compass operation, the factor parameter will be adaptively updated in each iteration, which can balance the ability between global and local search. In the final landmark operation, the gradual decreasing pigeon population-updating strategy is introduced to prevent premature convergence and local optimum. Finally, the demonstration of the proposed algorithm on UAV path planning problem is presented, and the comparison result indicates that the performance of our algorithm is better than that of particle swarm optimization (PSO), pigeon-inspired optimization (PIO), and its variants, in terms of convergence and accuracy.

Keywords: pigeon-inspired optimization (PIO); unmanned aerial vehicle (UAV); path planning; quantum behavior; adaptive operator

1. Introduction

Unmanned aerial vehicles (UAVs) have been widely and more extensively applied for various practical military and civilian missions, e.g., disaster circumstances, modern warfare, and weather monitoring in recent years, which benefit from their strong viability, low cost, and excellent maneuverability [1,2]. Autonomous navigation and guidance are the most important requirements of the UAV system, of which the basic component is path planning. Generally, path planning aims not only to generate a trajectory to a target in avoiding obstacles or collisions (assuming reference flight conditions and providing maps of the environment), but also to optimize a given function under kinematic and/or dynamic constraints [3]. The classical UAV path planning problem can be modeled as a nonlinear optimization problem aiming to find a flyable path from a given start position to a target position [4]. There does not exist an algorithm that provides an exact analytic solution to such a problem. A series of methods have been proposed and widely applied to solve the optimization problem, such as simulated annealing algorithm [5], artificial potential field algorithm [6], A* algorithm [7], RRT algorithm [8], polynomial optimization method [9], and heuristic approaches [10].

In recent decades, inspired by the organized behavior of natural biological groups, numerous swarm-intelligence optimization algorithms have been proposed to be applied to UAV path planning problem [11,12]. Notable examples include ant colony optimization algorithm (ACO) [13], particle swarm optimization algorithm (PSO) [14], fruit fly optimization algorithm (FOA) [15], and pigeon-inspired optimization algorithm (PIO) [16]. Various merits, including simple structure, general problem adaptability, and rapid search rate, make swam-intelligence optimization algorithms a promising tool for solving UAV path planning problems, especially under varied and complex environments.

The ACO algorithm can imitate the behavior of pheromone-mediated information transferred among ant colonies to optimize path planning problems [17]. However, ACO usually incurs a large number of calculations, and it is easy to fall into the local optimum. To address these problems, direction guidance and chaotic theory were applied to ACO for efficiency improvement [18,19]. By adding the chaos disturbance factor and direction guiding factor into the ACO, the direction of the search is strengthened, and the ACO can jump out of the local optimum. An improved ACO with redefined heuristic information function and pheromone updating method was applied to multi-UAV path planning to generate initial path [20]. With the redefined updating method, the probability of falling into local optimum and the global search ability are improved.

Besides, fruit fly behavior-inspired algorithm, named FOA, has been successfully applied to solve optimal path planning problems. Similar to the other swarm-intelligence algorithms, the conventional FOA still has the drawback that it easily falls into a local optimum. In order to improve the search efficiency and global search ability of the conventional FOA, Zhang et al. proposed a novel angle-encoded FOA with mutation adaptation mechanism (θ-MAFOA) and adopted the novel algorithm to the UAV path planning problem [21]. With the mutation adaption mechanism, the ability of exploitation and exploration of FOA are balanced.

PSO is a random search algorithm and has been widely applied to UAV path planning [22]. To avoid falling into the local optimum, integrating the PSO with other intelligence algorithms becomes even more efficient. Vincent made a comparison of parallel genetic algorithm and PSO for real-time UAV path planning to avoid poor solutions [23]. The generated UAV path is composed of vertical helices, line segments, and circular arcs, and a multi-objective cost function was established to evaluate the characteristic of the path by using this planning method. A hybrid differential evolution (DE) and quantum-behaved particle swarm optimization (DEQPSO) were used to plan the route for UAV on the sea [24]. Optimizing the design parameters is also another effective method for performance improvement. With the hybrid PSO strategy [25], the accuracy and the convergence rate of UAV path planning can be further increased by adding a contraction factor. Asma Ayari also proposed another improved PSO algorithm that satisfies Gaussian dynamic distribution and reduces the randomness of particles [26].

PIO is a novel swarm-intelligence optimization algorithm which is inspired by the behavior of pigeon homing with the help of landmark, map, and compass [27]. The PIO algorithm is composed of two operators, i.e., map and compass operator and landmark operator. Since PIO has demonstrated its fast convergence rate and adaptability in nonlinear optimizations, it has been widely applied to UAV path planning problems [28], multiple spacecraft position problem [29], and multi-UAV cooperation control [30,31]. In order to further improve the convergence rate and avoid the premature convergence problem, new biological behavior characteristics were introduced into PIO. Some predatory-prey pigeon-inspired optimization (PPPIO) algorithms were proposed and applied to UAV path planning in 3D static and dynamic environment [32,33]. The most recent hybrid pigeon-inspired optimization with quantum theory (QPIO) was proposed to solve the premature convergence problem and continuous optimization problems [34]. However, the existing PIO algorithm and its variants did not perform initial route optimization process. PPPIO, along with QPIO, failed to take account of global search and local search synchronously. Moreover, the path generated by QPIO is not smooth, and the existing problem of population decline too fast in landmark operator.

To address the above shortcomings, we introduce some adaptive operators into conventional QPIO to improve the capability of searching appropriate solutions. The chaotic strategy is introduced in our algorithm to generate some initial solutions with wider solution space coverage. Differing from other conventional swarm-intelligence algorithms, the map and compass factor R, varying with the number of iterations, is adopted to balance global search ability and local search ability in our algorithm. At the landmark operator, a new population updating equation is proposed to increase the search diversity. Simulation results and comparisons verify the feasibility and effectiveness of our proposed algorithm.

The rest of the paper is organized as follows: Section 2 describes the preliminaries, including the problem formulation of UAV path planning and the theory of QPIO. Section 3 shows the implementation procedure of UAV path planning using our proposed adaptive operator quantum-behaved pigeon-inspired optimization (AOQPIO). Then, we compare the quality of the path produced by PIO, PSO, QPIO, and AOQPIO in Section 4. Section 5 is the conclusion.

2. Preliminaries

2.1. Problem Formulation

UAV needs to plan a safe and short flying path from starting position to target position. However, numerous obstacles and no-fly zones fulfill complex environments, such as urban, indoor, or forest area. It is crucial to take the constraints of nonlinear dynamics and unstructured environment into account. Further, the fly speed and altitude changing need to be also considered. In the paper, we model the above planning problems as a mixed objective optimization problem with constraints on UAV dynamic and threats, and assume that UAVs satisfy following dynamics.

$$\begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} q\cos\phi - r\sin\phi \\ p + (r + \cos\phi + q\sin\phi)\tan\theta \\ \frac{1}{\cos\theta}(r\cos\phi + q\sin\phi) \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} u\cos\theta\cos\psi + v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) + w(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi) \\ u\cos\theta\cos\psi + v(\sin\phi\sin\theta\sin\psi + \cos\phi\sin\psi) + w(-\sin\phi\sin\psi + \cos\phi\sin\theta\sin\psi) \\ u\sin\theta - v\sin\theta\cos\theta - w\cos\phi\cos\theta \end{bmatrix} \tag{2}$$

where $\eta = \begin{bmatrix} \theta & \phi & \psi \end{bmatrix}$ are Euler angles; and $\Omega = \begin{bmatrix} p & q & r \end{bmatrix}$ are angular velocities; $X = \begin{bmatrix} x & y & z \end{bmatrix}$ and $V = \begin{bmatrix} u & v & w \end{bmatrix}$ are position and linear velocities respectively in the North-East-Down inertial reference.

Further, the threats in the environment are simplified as cylinders that denoted by distinct vectors, including position coordinates (x_k, y_k, z_k), radius r_k, and corresponding threat levels t_k. Threat level t_k is used to indicate the impact degree of threats on the UAV path.

By taking fly time, path length, and security requirements into account, the UAV path planning problem can be established as the following optimization problem:

$$\min J = c_1 J_l + c_2 J_t + c_3 J_a c_1 + c_2 + c_3 = 1 \tag{3}$$

The cost function J is consisted of J_l, J_a, and J_t, which represent path length cost, the altitude cost and the threat cost, respectively. The path length cost and altitude cost can be directly obtained by

$$J_l = \sum_{i=1}^{n} l_i^2, \tag{4}$$

$$J_a = \sum_{i=1}^{n} h_i, \tag{5}$$

where n is the amount of path segments; l_i and h_i are the length and the average altitude above the sea level of the i-th path segment, respectively. To simplify the total threat cost model, an efficient approximation is adopted to the exact solution. In our work, threat cost, which can be obtained by five points, of each edge connects two discrete points along the edge [16].

$$J_t = \frac{L_{ij}}{5}\sum_{k=1}^{N_t} t_k\left(\frac{1}{d_{0.1,k}^4}+\frac{1}{d_{0.3,k}^4}+\frac{1}{d_{0.5,k}^4}+\frac{1}{d_{0.7,k}^4}+\frac{1}{d_{0.9,k}^4}\right), \tag{6}$$

where $d_{0.1,k}$ is the length between the 1/10 point and the k-th threat center. We divide the line segment between two discrete points $(x_{i-1}, y_{i-1}, z_{i-1})$, (x_i, y_i, z_i) into five segments of the same length, and select five of them. The threat cost of the line segment between two discrete points is approximately equal to the total cost of the selected five points. Threat computation model is shown in Figure 1.

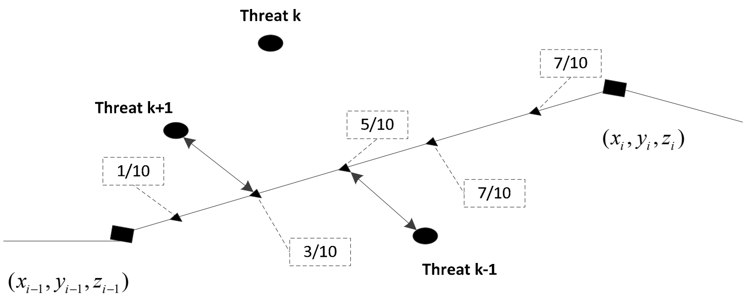


Figure 1. The image of threat computation model.

2.2. QPIO

QPIO, which consists of two operators—the quantum behavior map and compass operator and the landmark operator, is an emerging variant of PIO algorithm. The quantum behavior is introduced to PIO to increase the local search capacity as well as the randomness of the position, so that the QPIO algorithm has the ability to improve the optimization efficiency and avoid premature convergence problem.

2.2.1. Map and Compass Operator with Quantum Behavior

Each pigeon has its position $X_i = \begin{bmatrix} X_{i,1} & X_{i,2} & \cdots & X_{i,D} \end{bmatrix}$ and speed $V_i = \begin{bmatrix} V_{i,1} & V_{i,2} & \cdots & V_{i,D} \end{bmatrix}$ in every feasible position of pigeon, where $i = 1,2,3\cdots N_p$. N_p, and D refers to the number of feasible solutions and their dimensions, respectively. In the D dimensional solution space, the position and speed of individuals are updated as follows:

$$V_i(t) = V_i(t-1) * e^{-Rt} + rand * (X_g - X_i(t-1)), \tag{7}$$

$$X_i(t) = X_i(t-1) + V_i(t), \tag{8}$$

where $V_i(t)$ and $X_i(t)$ are the speed and position of the pigeon i at iteration t; X_g represents the best position that the whole pigeon obtained by comparing the positions of all the pigeons until iteration $t-1$; $rand \in (0,1)$ is a randomly generated number; $R \in (0,1)$ refers to the map and compass factor. Importantly, the map and compass factor R is a constant value that plays an important role in convergence of the algorithm. The smaller R becomes, the larger value of e^{-Rt} becomes. Therefore, pigeons inherit a greater speed, which is conducive to fast convergence and better global search ability; conversely, large R would lead to only local search.

In order to improve the optimization efficiency and avoid premature convergence problem, quantum-behaved theory is added to the conventional PIO (7)–(8) [35,36].

$$X_i(t) = \begin{cases} P_i(t) + \varepsilon * |m\_best_i(t) - X_i(t-1)| * \ln(1/m) * e^{-Rt} & \varphi > 0.5 \\ P_i(t) - \varepsilon * |m\_best_i(t) - X_i(t-1)| * \ln(1/m) * e^{-Rt} & \varphi < 0.5 \end{cases}, \tag{9}$$

$$P_i(t) = \varphi * X_{pi}(t) + (1-\varphi) * X_g, \tag{10}$$

$$m\_best_i(t) = \tfrac{1}{N}\sum_{i=1}^{N} X_{pi}(t), \tag{11}$$

where m_best_i is the mean value of best pigeon positions, and $X_{pi}(t)$ is the local best solution that pigeon i has obtained before the iteration t; ε is the expansion-contraction coefficient:

$$\varepsilon = 1 - q * (t/N_{c1\max}), \tag{12}$$

where q is an experimentally defined parameter; $N_{c1\max}$ refers to the maximum number of generation; φ and m are random numbers in the given interval $[0,\ 1]$.

2.2.2. Landmark Operator

In each iteration loop, the landmark operator will sort fitness values of the current individual, and round off each half after discarding the poor quality individuals. The position $X_c(t)$ of remaining pigeon center is used as a landmark and the reference direction of the flight. The update formula is as follows:

$$N_p(t) = N_p(t-1)/2, \tag{13}$$

$$X_c(t) = \frac{\sum X_i(t) * fitness(X_i(t))}{N_p \sum fitness(X_i(t))}, \tag{14}$$

$$X_i(t) = X_i(t) + rand * (X_c(t) - X_i(t)), \tag{15}$$

In the Equations (13) and (14), N_p is the population of pigeons and $fitness(X_i(t))$ is the quality of the pigeon individual which is constructed by the cost function according to Equations (3)–(6).

Remark. *For the minimum optimization problems such as the path planning problem, we can choose* $fitness(X_i(t)) = \frac{1}{J+eps}$, *where* J *is the optimization function described in Equation (3) and eps refers to the relative accuracy of the float point.*

3. Method

In this section, the population initialization process of QPIO is optimized by logistic mapping and the map and compass factor R is modified to balance the global and local search ability. In the landmark operator, new population updating and pigeon position X_i updating strategies are given.

3.1. Initialization Process

QPIO presents the best convergence rate among all PIO algorithms, but it is still suffering from a major disadvantage that is vulnerable to fall into local optimum [37]. Specifically, when the initialization is away from the global optimum, this error will guide other pigeons to search in wrong spaces and even leading, eventually, to failure. Since most initialization of pigeon population is achieved by random distribution, their performance quality could not be ensured in conventional methods. Therefore, the chaotic strategy is introduced in our algorithm to generate some initial solutions that can obtain wider solution space coverage.

Chaos is a kind of seemingly random motion appearing in the decisive dynamic system. Chaotic motion has three distinct characteristics: (1) high sensitivity to initial values; (2) ergodicity of motion trajectories; (3) randomness. These properties of chaos can be used to initialize the pigeon with chaotic variables. Logistic mapping is a common method used in this paper to produce chaotic variables, and avoid premature convergence [38]. In the initialization process, the logistic mapping strategy is used to make the pigeon position random and traversable, so that it has the opportunity to jump out of the local optimum and find the global optimum, thus effectively avoiding premature convergence. The D dimensions vector $u_1 = (u_{1,1}, u_{1,2} \ldots u_{1,D}) \in [0,1]^D$ is generated randomly. Then, a chaotic sequence U with N vectors $u_1, u_2, \ldots u_N$ is obtained according to the recurrence Equation (16):

$$u_{n+1} = \mu * u_n(1 - u_n),\ n = 0,1,2,\cdots N-1\ 0 < u_1 < 1. \tag{16}$$

After the chaos process, the chaos sequence U is converted into the problem solution space $[X_{min}, X_{max}]^D$. The corresponding individual initialization formula can be converted as

$$X_{i,j} = X_{min,j} + u_n * (X_{max,j} - X_{min,j}),\ j = 1, 2 \cdots D. \tag{17}$$

Initialization of pigeon by chaotic sequence is based on the generation of initial values, from which the initial population can be determined by selecting the best one. Although it does not change the stochastic nature of AOQPIO initialization, the chaotic process improves the diversity of population and the ergodicity of pigeon search by applying chaotic theory.

3.2. *Adaptive Map and Compass Factor Strategy*

The selection of parameters significantly has an impact on the result. Conventional methods of choosing appropriate parameters can be roughly divided into two categories [39]: parameter adjustment and parameter control. Parameter adjustment is used for parameters preset before the beginning, and the parameters that are kept constant throughout the execution of the algorithm. Parameter control means that parameters run with the initial value at the beginning of the algorithm. Then, they can be adaptively updated according to the operation of the algorithm.

QPIO is simple to be implemented as its parameter number is limited. However, it still poses a great challenge to adjusting the parameters in QPIO. As mentioned in a previous section, the turning of map and compass factor R can adjust the algorithm on better global or local searching capability. If R is set to some constant values, the algorithm is not capable of coordinating the global and local search synchronously during the optimization procedure. Thus, we aim to find a function with the following ideal characteristics. The initial value of R is small with gradual changes; as the independent variable increases, it rapidly reaches the preset values [40].

$$R(\tau) = \frac{1}{a + b * e^{-\tau}} \tag{18}$$

$$\tau = -10 + \frac{N_c * 20}{N_{c1\max}} \tag{19}$$

Apparently, $R(\tau)$ belongs to $[0, 1/a]$; $\tau \in [-10, 10]$ is determined by the number of iteration and its maximum; N_c is the number of iteration; parameter b can adjust the rising speed. Specifically, the value of R is coincident with conventional QPIO algorithm when the value of a is set to 1.

By using the dynamic parameters tuning strategy (18) and (19), R is small at the beginning and it gradually increases after fully global search. The local search is better completed accordingly. With the consideration on the value range of R and the balance of global-local search, we set $a = 1$ and $b = 100$ in this paper.

3.3. *Adaptive Compression Factor Strategy*

The number of pigeons decreases by half after each iteration in a conventional QPIO algorithm, according to Equation (14). However, if the pigeon number decreases too fast, only one pigeon can survive after a small amount of iterations eventually, which prevents globally searching for the algorithm. The optimization performance is reduced in the later searching stage. Besides, the landmark operator in the conventional QPIO do not consider the waypoints distribution, which will influence the smoothness of the planning path.

To address the above problem, we propose a new pigeon number updating strategy in the landmark operator.

$$N_p(t) = w * N_{P_{max}} - \mathrm{N}_{dec}, \tag{20}$$

where $w \in [0, 1]$ is constant and N_{dec} is a constant parameter in the initial. The new formula of pigeon position updating in the landmark operator is shown, as below.

$$\omega = s + (1 - s) * \cos(t * \pi / N_{c2max}), \tag{21}$$

$$X_i(t) = X_i(t) + rand * \omega * (X_c(t) - X_i(t)), \tag{22}$$

where $s \in (0, 1)$ is a constant and N_{c2max} is the maximum number of generation after the landmark operation finished.

3.4. Procedures of AOQPIO

Differing from other conventional PIO algorithms, the process of the proposed method in this paper has mainly three improvements. Firstly, the chaotic strategy is introduced in our algorithm to generate some initial solutions with wider solution space coverage. Further, the map and compass factor R, varying with the number of iterations, is adopted to balance global search ability and local search ability. Last but not least, the new population updating equation is proposed to increase the search diversity at the landmark operator period. Overall, the complete procedure of AOQPIO algorithm can be presented as the following Algorithm 1:

Algorithm 1: AOQPIO Algorithm.

```
      /*Initialization*/
1     Set initial values for $D$, $N_p$, $N_{c1\max}$, $N_{c2\max}$, $q$, $w$, $N_{dec}$;
2     Generate the chaos sequence $U$ according to Equation (20);
3     Set initial path $X_i = [\ X_{i,1} \quad X_{i,2} \quad \cdots \quad X_{i,D}\ ]$, velocity $V_i = [\ V_{i,1} \quad V_{i,2} \quad \cdots \quad V_{i,D}\ ]$ for pigeon
      according to Equation (21);
4     Calculate fitness values of different pigeon individuals, Set $X_{pi} = X_i$,
      $X_g = \text{argmin}[calculate\_fitness(X_{pi})]$;
      /*Map and compass operations with quantum-behavior*/
5     for $N_c$ = 1:$N_{c1\max}$
6           Generate parameter $\varphi$, $\varepsilon$, $m$;
            Update parameter $R$ according to Equations (22) and (23);
8           Sum the $X_{pi}(i)$ for each $i$;
10          Calculate m_best according to Equation (15);
11          for $i$ = 1:$N_p$
12             Calculate $P_i$ according to Equation (14);
13             Update $V_i$ according to Equation (11);
14             Update $X_i$ with quantum-behavior according to Equation (10);
15             new_fitness = calculate_fitness($X_i(N_c)$);
16             If new_fitness < Fitness($i$) then Fitness($i$) = new_fitness; $X_{pi}$=$X_i$;
18             end if
19              $X_g = \text{argmin}[calculate\_fitness(X_{pi}(N_c))]$;
20          end for
21    end for
22    /*Landmark operator*/
24    for $N_c$ = $N_{c1\max}$+1:$N_{c2\max}$
25         Rank all the available pigeon individuals according to their fitness values;
26         Update the population of pigeon $N_p$ according to Equation (24);
27         Calculate the center of the pigeons $X_c(N_c)$ according to Equation (18);
28         for $i$ = 1:$N_p$
29            Update $X_i$ according to Equation (26);
30            Evaluate $X_i$, and update $X_{pi}$ and $X_g$ according to line (16)–(20);
31         end for
32    end for
33    /*Output*/
34    $X_g$ is output as the global optimal of the fitness function
```

3.5. AOQPIO-Based UAV Path Planning

In order to further illustrate AOQPIO, in this paper, to solve UAV path planning problem, the detailed steps of AOQPIO-based UAV path planning are described in this section. If the user-defined number of control points of the UAV path is D, then the path planner should determine $D1 = 3 * D$ coordinate parameters, namely $X_1, X_2, \cdots, X_{D1}$, where $(X_i, X_{D+i}, X_{2D+i}), i = 1, 2, \cdots, D$ denote the three dimensional coordinates in solution space. The point (x_i, x_{D+i}, x_{2D+i}) in solution space can be transformed into the point $(x(i), y(i), z(i))$ in the real flying space by the following transform equation [41]:

$$\begin{bmatrix} x(i) \\ y(i) \\ z(i) \end{bmatrix} = \begin{bmatrix} \cos\xi & \sin\xi & 0 \\ -\sin\xi & \cos\xi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ x_{D+i} \\ x_{2D+i} \end{bmatrix} + \begin{bmatrix} x_{start} \\ y_{start} \\ z_{start} \end{bmatrix}, \tag{23}$$

where the $(x_{start}, y_{start}, z_{start})$ represents the coordinate of the given start point in the flying space, and the ξ is the angle that the x-axis in solution space counterclockwise rotate to parallel segment start and goal point. The implementation procedure of AOQPIO algorithm applied to UAV path planning can be described as follows:

Step 1: Environmental modeling, initialize terrain and threat information, including the center coordinates of threats, radius, and threat level of the threat;

Step 2: According the environment model, build the path planning optimization function on the basis of Equations (3)–(6), and initialize the detailed information about the path planning task;

Step 3: The population information and algorithm parameters are initialized, including the population size N_p, the number of waypoints D, the solution dimension $D1$, the operation operator parameter, and the number of iteration $N_{c1\max}$ and $N_{c2\max}$ for two operators, and $N_{c2\max} > N_{c1\max}$;

Step 4: Initialize individual velocity and position information by chaos sequence method according to Equations (16) and (17). Compare the fitness value $fitness()$, which is defined based on the optimization function, of each pigeon and find the current best position of pigeons;

Step 5: Operate the map and compass operator. Firstly, we update the parameter R according to Equations (18) and (19). Next, we update the velocity and position of each pigeon using quantum-behaved method. Then, we compare all the pigeons' fitness value and update the new global best position X_p;

Step 6: If the number of iterations $N_c > N_{c\max1}$, the iteration switches from the map compass operator to the landmark operator, otherwise, it returns to the step 5;

Step 7: Operate the landmark operator. According to the adaptation value, update the number of pigeons by Equation (20), then calculate the center of the pigeon according to Equation (14) and adjust the position of each pigeon to fly to the center of the pigeon with adaptive compression factor strategy according to the Equations (21) and (22);

Step 8: If the number of iterations $N_c > N_{c\max2}$, the iteration terminates, and the result is output, otherwise, it returns to the step 7.

Step 9: Transform the best solution result into the waypoint in real flying space according to Equation (24).

To illustrate this procedure further, the pseudocode of this procedure is also given as the following Algorithm 2.

Algorithm 2: AOQPIO Based path planner.

```
        /*Environment modeling*/
1   Initialize terrain and threat information, including:
2   The mathematical form of terrain in Equation (28);
3   The center coordinates of threats, radius and threat level of the threats in Table 1;
        /*Problem modeling*/
    Build the path planning optimization function on the basis of Equation (3)
        /*Initialization*/
4     Set initial values for $D = 30$, $D1 = 3 * D = 90$, $N_p = 150$, $N_{c1\max} = 150$, $N_{c2\max} = 200$, $q = 0.5$, $w = 0.8$,
    $N_{dec} = 10$;
5     Generate the chaos sequence $U$ according to Equation (20);
6     Set initial value $X_i = [\ X_{i,1}\quad X_{i,2}\quad \cdots\quad X_{i,D1}\ ]$, $V_i = [\ V_{i,1}\quad V_{i,2}\quad \cdots\quad V_{i,D1}\ ]$ for pigeon according
    to Equation (21);
7     Divide the $X_i$ into $[\ X_{i,1}\quad X_{i,2}\quad \cdots\quad X_{i,D}\ ]$, $[\ X_{i,D1+1}\quad X_{i,D+2}\quad \cdots\quad X_{i,2D}\ ]$,
    $[\ X_{i,2D+1}\quad X_{i,2D+2}\quad \cdots\quad X_{i,D1}\ ]$, the same to $V_i$
8     Calculate fitness values of different pigeon individuals with cost function, using the real coordinate
    transformed by the position value in solution space according to Equation (27);
9     Set $X_{pi} = X_i$, $X_g = \mathrm{argmin}[calculate\_fitness(X_{pi})]$;
      /*Map and compass operations with quantum-behavior*/
10    for $N_c$ = 1:$N_{c1\max}$
11            Generate parameter $\varphi$, $\varepsilon$, $m$;
12            Update parameter $R$ according to Equations (22) and (23);
13            Sum the $X_{pi}(i)$ for each $i$;
14            Calculate m_best according to Equation (15);
15            for $i$ = 1:$N_p$
16                Calculate $P_i$ according to Equation (14);
17                Update $V_i$ according to Equation (11);
18                Update $X_i$ with quantum-behavior according to Equation (10);
19                new_fitness = calculate_fitness ($X_i(N_c)$);
20                If new_fitness < Fitness($i$) then Fitness($i$) = new_fitness; $X_{pi}$=$X_i$;
21                end if
22                $X_g = \mathrm{argmin}[calculate\_fitness(X_{pi}(N_c))]$;
23            end for
24      end for
25    /*Landmark operator*/
26      for $N_c$ = $N_{c1\max}$+1:$N_{c2\max}$
27            Rank all the available pigeon individuals according to their fitness values;
28            Update the population of pigeon $N_p$ according to Equation (24);
29            Calculate the center of the pigeons $X_c(N_c)$ according to Equation (18);
30            for $i$ = 1:$N_p$
31                Update $X_i$ according to Equation (26);
32                Evaluate $X_i$, and update $X_{pi}$ and $X_g$ according to line (16)–(20);
33            end for
34      end for
35    /*Output*/
36    $X_g$ is output as the global optimal of the fitness function
37    Transform the best solution result into the coordinate of point in real flying space according to Equation (27)
```

Table 1. The parameters used in experiments of different algorithms.

Parameters	AOQPIO	PIO	QPIO	PSO
N_p	150	150	150	150
D	30	30	30	30
$N_{c1\max}$	150	150	150	200
$N_{c2\max}$	200	200	200	——
w	0.8	——	——	——
s	0.5	——	——	——
a	1	——	——	——
b	100	——	——	——
q	0.5	——	0.5	——

4. Results and Discussion

To validate the efficiency of AOQPIO algorithm on UAV path planning problem, the optimization results in 3D environment are presented. The 3D environment containing a complex undulating terrain as shown in Figure 2, can also be found in [42]:

$$z(x,y) = \left|\sin(x/5+1) + \sin(y/5) + \cos\left(\alpha * \sqrt{x^2+y^2}\right) + \sin(\beta * \sqrt{x^2+y^2})\right|, \tag{24}$$

where α, β are constant parameters, and z represents the altitude of a certain point.

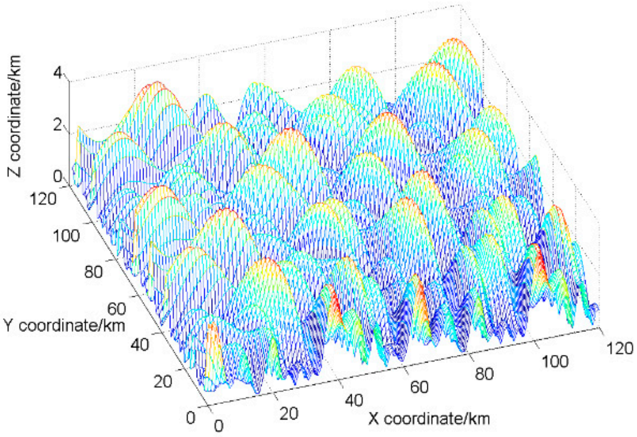
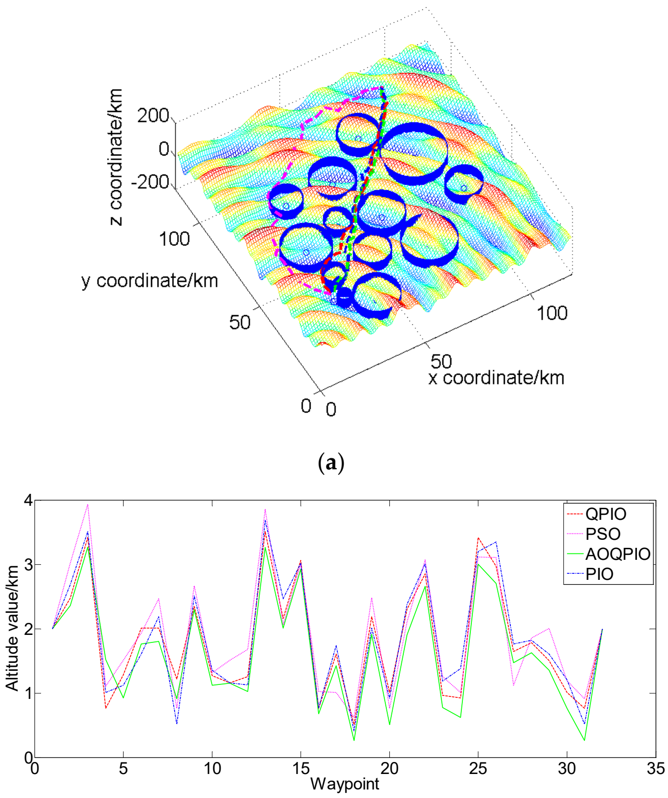


Figure 2. The image of the 3D undulating terrain environment.

Compared with existing swarm-intelligence algorithms, i.e., PIO, QPIO, PSO, we utilized path distance, cost, and altitude variations to evaluate the performance of above algorithms. All of the experiments were conducted by MATLAB 2014a on a PC with 2.4GHz CPU. The corresponding parameters used in the algorithms are presented in following Table 1.

To verify the generalization ability of the proposed algorithm in different environments, experiments, which were utilized by the mentioned algorithms, were carried out in three maps with different threat sources. The corresponding parameters of the threat sources in the three maps are provided as supplementary in the enclosure. The start coordinates of the planning mission were (18,20,2) and the target coordinates were (90,105,2).

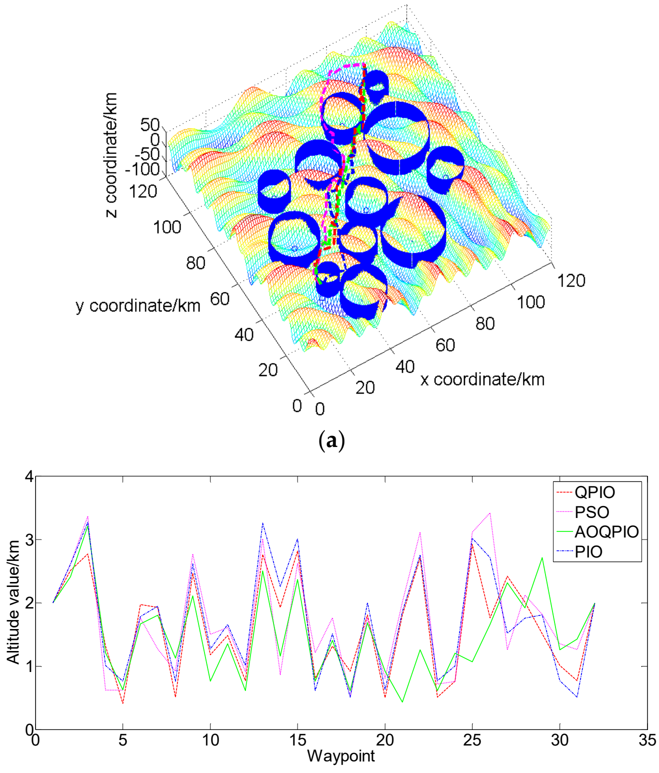
Apparently, the planning results calculated by AOQPIO, indicated with a green line in Figures 3a, 4a and 5a, are smoother than the results of other algorithms. Convergence curves of the cost values in the three scenarios are displayed in Figures 6–8. According to the altitude value in Figures 3b, 4b and 5b, most of the waypoints generated by our algorithm are lower than those of other algorithms. Further, according to the convergence results of cost function value, our algorithm has the ability of converging more stably and quickly when compared with other three algorithms.



(**a**)

(**b**)

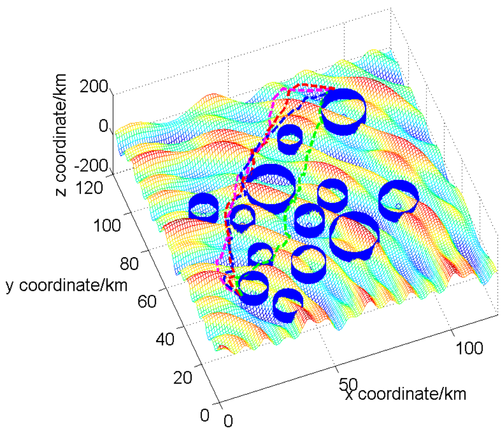
Figure 3. (**a**) is the result of all algorithms and (**b**) is the result of optimal unmanned aerial vehicle (UAV) altitude of the experiment carried out in map 2.



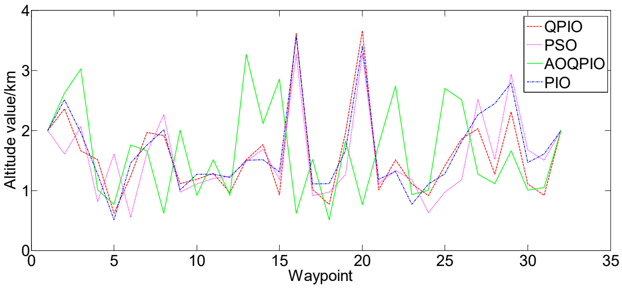
(**a**)

(**b**)

Figure 4. (**a**) is the result of all algorithms and (**b**) is the result of optimal UAV altitude of the experiment carried out in map 2.



(**a**)



(**b**)

Figure 5. (**a**) is the result of the experiment and (**b**) is the result of optimal UAV altitude of the experiment carried out in map 3.

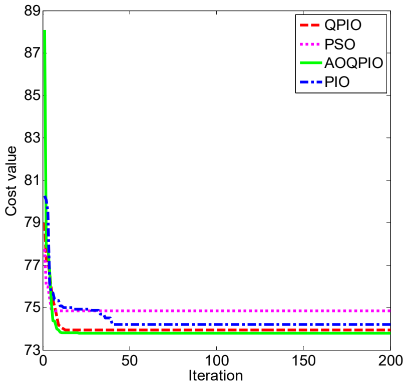


Figure 6. The comparative curves of iteration of the algorithms of the experiment carried out in map 1.

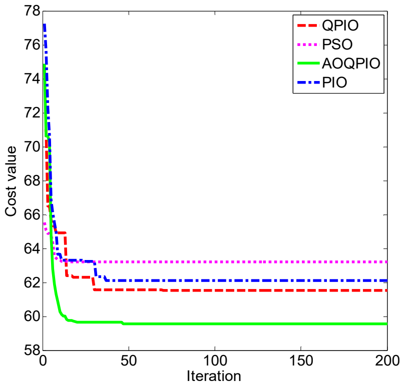


Figure 7. The comparative curves of iteration of the algorithms of the experiment carried out in map 2.

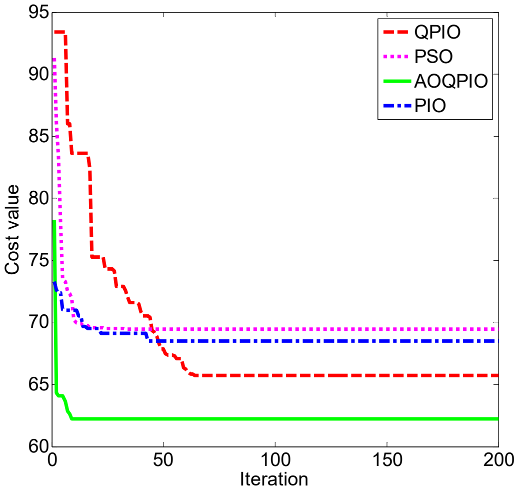


Figure 8. The comparative curves of iteration of the algorithms of the experiment carried out in map 3.

The curve of the cost iteration in map 1 shows that our algorithm convergences at the eighth iteration and the convergence value is 73.8954, which is lower than the convergence value of other algorithms. Similar experiment results obtained on the above three test scenarios verify that the performance of AOQPIO is independent of changed battlefield environments that mainly mean different enemy threats. Clearly, the modifications provided in this paper had a positive effect on AOQPIO.

To avoid the influence of random initial values, another 40 experiments in each map were carried out and the average values of path length, cost, and searching time were calculated. The statistical results are listed in Table 2. From the results in map 1, the path generated by AOQPIO is 5.9% lower than those of other algorithms, and cost decreases by at least 0.4% separately. Similar to the experiment carried out in map 1, our algorithm also has the best efficiency and can generate the shortest path in the experiments carried out in map 2 and map 3. Comprehensively comparing the statistical data of all algorithms on the three maps, AOQPIO still shows its superiority to other algorithms in terms of searching ability and efficiency.

Table 2. The statistical results of path length, cost, and running time of the algorithms in the three maps.

Algorithm	**Path Length(km)**			**Cost**			**Searching Time (ms)**		
	Map1	**Map2**	**Map3**	**Map1**	**Map2**	**Map3**	**Map1**	**Map2**	**Map3**
AOQPIO	155.4329	154.3672	152.2775	78.8326	58.2366	63.1964	2755	2632	2553
PIO	177.4352	171.2873	165.4109	79.9846	61.3127	65.3127	3127	2607	2515
QPIO	165.2263	160.4521	169.5543	79.1623	63.0571	65.4218	2743	2613	2538
PSO	170.5448	168.6482	170.3458	79.5647	63.8452	68.8647	2716	2596	2491

From the experiment results showed above, we can see that AOQPIO is significantly superior to other swarm-intelligence algorithms tested in this paper. The chaotic strategy indeed can improve the global search ability of the basic QPIO. Furthermore, the adaptive operator strategies help the AOQPIO to search more fully and carefully. With the adaptive compression factor, a smoother path can be generated by AOQPIO. In a word, the proposed AOQPIO has the advantage over the conventional QPIO, as well as PSO and PIO algorithms in terms of searching ability, stability, and robustness.

5. Conclusions

The AOQPIO algorithm for UAV path planning in 3D undulating terrain was proposed in this paper. Various improvements have been incorporated into conventional QPIO algorithms to ensure the convergence rate and avoid local optimum. By utilizing logistic mapping method, the introduced pigeon population initialization procedure improved the diversity of population and the ergodicity of pigeon, which can accelerate overall convergence rate. Further, the adaptive map and compass factor is allowed to vary in each iteration during the process of AOQPIO algorithm, which ensures a balanced global and local search ability. To prevent the fast decrease and algorithm premature convergence at some local optimum, a new population updating strategy was presented regarding these issues. Simulation and comparison results showed that our algorithm has a better convergence rate, and the UAV path obtained by our algorithm has the lowest cost among all existing algorithms.

Author Contributions: Project administration, J.Z.; software, Y.X.; supervision, C.H.; writing—original draft, Y.X.; writing—review & editing, C.H.

Funding: This work was funded by Fundamental Research Funds for the Central Universities (Grant No. BLX201605, No. 2016ZCQ08), and National Natural Science Foundation of China (Grant No. 61703047).

Acknowledgments: The authors would like to thank the editors and reviewers for their helpful suggestions and comments, which indeed improved the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tisdale, J.; Kim, Z.; Hedrick, J.K. Autonomous UAV path planning and estimation. *IEEE Robot. Autom. Mag.* **2009**, *61*, 35–42. [CrossRef]
2. Trotta, A.; D'Andreagiovanni, F.; Felice, M.; Natalizio, E.; Chowdhury, K. When UAVs Ride A Bus: Towards Energy-efficient City-scale Video Surveillance. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications, Honolulu, HI, USA, 15–19 April 2018.
3. Zhang, Y.; Li, S. Distributed Biased Min-Consensus with Applications to Shortest Path Planning. *IEEE Trans. Autom. Control* **2017**, *62*, 5429–5436. [CrossRef]
4. Otto, A.; Agatz, N.; Cambell, J.; Golden, B.; Pesch, E. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks* **2018**, 1–48. [CrossRef]
5. Ho, Y.J.; Liu, J.S. Simulated annealing based algorithm for smooth robot path planning with different kinematic constraints. In Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, 22–26 March 2010.
6. Sheng, J.; He, G.; Guo, W.; Li, J.H. An Improved Artificial Potential Field Algorithm for Virtual Human Path Planning. In Proceedings of the International Conference on Entertainment for Education Digital Techniques & Systems, Changchun, China, 16–18 August 2010.
7. Jeddisaravi, K.; Alitappeh, R.J.; Guimarães, F.G. Multi-objective mobile robot path planning based on A* search. In Proceedings of the International Conference on Computer & Knowledge Engineering, Mashhad, Iran, 20 October 2017.
8. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *arXiv*, 2011; arXiv:1105.1186.
9. Richter, C.; Bry, A.; Roy, N. *Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments*; Springer: Cham, Switzerland, 2016.
10. Thoa Mac, T.; Copot, C.; Tran, D.T.; Keyser, R.D. Heuristic approaches in robot path planning: A survey. *Robot. Auton. Syst.* **2016**, *86*, 13–28.
11. Yang, P.; Tang, K.; Lozano, J.A.; Gao, X. Path Planning for Single Unmanned Aerial Vehicle by Separately Evolving Waypoints. *IEEE Trans. Robot.* **2015**, *31*, 1130–1146. [CrossRef]
12. Das, P.K.; Behera, H.S.; Panigrahi, B.K. A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm Evol. Comput.* **2016**, *28*, 14–28. [CrossRef]
13. Lee, J. Heterogeneous-ants-based path planner for global path planning of mobile robot applications. *Int. J. Control Autom. Syst.* **2017**, *15*, 1–16. [CrossRef]
14. Das, P.K.; Behera, H.S.; Das, S.; Tripathy, H.K.; Panigrahi, B.K.; Pradhan, S.K. A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment. *Neurocomputing* **2016**, *207*, 735–753. [CrossRef]
15. Pan, W.T. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. *Knowl. Based Syst.* **2012**, *26*, 69–74. [CrossRef]

16. Duan, H.; Ye, F. Progresses in Pigeon-inspired Optimization Algorithms. *J. Beijing Univ. Technol.* **2017**, *43*, 1–7.
17. Wen, Y.E.; Deng-Wu, M.A.; Fan, H.D. Algorithm for Low Altitude Penetration Aircraft Path Planning with Improved Ant Colony Algorithm. *Chin. J. Aeronaut.* **2005**, *18*, 18–23.
18. Cheng, J.; Miao, Z.; Li, B.; Xu, W. An improved ACO algorithm for mobile robot path planning. In Proceedings of the IEEE International Conference on Information & Automation, Ningbo, China, 1–3 August 2017.
19. Zhang, D.; Xian, Y.; Li, J.; Lei, G.; Chang, Y. UAV Path Planning Based on Chaos Ant Colony Algorithm. In Proceedings of the International Conference on Computer Science and Mechanical Automation, Hangzhou, China, 23–25 October 2015; pp. 81–85.
20. Huang, L.; Qu, H.; Ji, P.; Liu, X.; Fan, Z. A novel coordinated path planning method using *k*-degree smoothing for multi-UAVs. *Appl. Soft Comput.* **2016**, *48*, 182–192. [CrossRef]
21. Zhang, X.; Lu, X.; Jia, S.; Li, X. A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning. *Appl. Soft Comput.* **2018**, *70*, 371–388. [CrossRef]
22. Masdari, M.; Salehi, F.; Jalali, M.; Bidaki, M. A Survey of PSO-Based Scheduling Algorithms in Cloud Computing. *J. Netw. Syst. Manag.* **2017**, *25*, 122–158. [CrossRef]
23. Roberge, V.; Tarbouchi, M.; Labonte, G. Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. *IEEE Trans. Ind. Inform.* **2012**, *9*, 132–141. [CrossRef]
24. Fu, Y.; Ding, M.; Zhou, C.; Hu, H. Route Planning for Unmanned Aerial Vehicle (UAV) on the Sea Using Hybrid Differential Evolution and Quantum-Behaved Particle Swarm Optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2013**, *43*, 1451–1465. [CrossRef]
25. Geng, Q.; Zhao, Z. A Kind of Route Planning Method for UAV Based on Improved PSO Algorithm. In Proceedings of the Chinese Control and Decision Conference, Guiyang, China, 25–27 May 2013; pp. 2328–2331.
26. Ayari, A.; Bouamama, S. Collision-free optimal paths for multiple robot systems using a new dynamic distributed particle swarm optimization algorithm. In Proceedings of the International Conference on Advanced Robotics, Hefei/Tai'an, China, 27–31 August 2017; pp. 493–497.
27. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [CrossRef]
28. Zhang, X.; Duan, H.; Yang, C. Pigeon-Inspired optimization approach to multiple UAVs formation reconfiguration controller design. In Proceedings of the Guidance, Navigation and Control Conference, Yantai, China, 8–10 August 2014; pp. 2707–2712.
29. Zhang, S.; Duan, H. Gaussian pigeon-inspired optimization approach to orbital spacecraft formation reconfiguration. *Chin. J. Aeronaut.* **2015**, *28*, 200–205. [CrossRef]
30. Ran, H.; Luo, D.; Duan, H. Multiple UAVs mission assignment based on modified Pigeon-inspired optimization algorithm. In Proceedings of the Guidance, Navigation and Control Conference, Yantai, China, 8–10 August 2014; pp. 2692–2697.
31. Dou, R.; Duan, H. Lévy flight based pigeon-inspired optimization for control parameters optimization in automatic carrier landing system. *Aerosp. Sci. Technol.* **2017**, *61*, 11–20. [CrossRef]
32. Zhang, B.; Duan, H. Predator-Prey Pigeon-Inspired Optimization for UAV Three-Dimensional Path Planning. In *Advances in Swarm Intelligence*; Springer: Cham, Switzerland, 2014; pp. 96–105.
33. Zhang, B.; Duan, H. Three-Dimensional Path Planning for Uninhabited Combat Aerial Vehicle Based on Predator-Prey Pigeon-Inspired Optimization in Dynamic Environment. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2017**, *14*, 97–107. [CrossRef]
34. Li, H.; Duan, H. Bloch quantum-behaved Pigeon-inspired optimization for continuous optimization problems. In Proceedings of the Guidance, Navigation & Control Conference, Yantai, China, 8–10 August 2014.
35. Liu, Z.; Sun, H.; Hu, H. Two Sub-swarms Quantum-Behaved Particle Swarm Optimization Algorithm Based on Exchange Strategy. In Proceedings of the Third International Symposium on Intelligent Information Technology & Security Informatics, Jinggangshan, China, 2–4 April 2010.
36. Sun, J.; Feng, B.; Xu, W. Particle swarm optimization with particles having quantum behavior. In Proceedings of the 2004 Congress on Evolutionary Computation, Portland, OR, USA, 19–23 June 2004; Volume 1, pp. 325–331.

37. Wang, Y.; Li, R.; Zhou, Y.; Yin, M. A path cost-based GRASP for minimum independent dominating set problem. *Neural Comput. Appl.* **2017**, *28*, 143–151. [CrossRef]
38. Himstedt, M.; Maehle, E. Online semantic mapping of logistic environments using RGB-D cameras. *Int. J. Adv. Robot. Syst.* **2017**. [CrossRef]
39. Eiben, A.E.; Hinterding, R.; Michalewicz, Z. Parameter Control in Evolutionary Algorithms. *IEEE Trans. Evol. Comput.* **1999**, *3*, 124–141. [CrossRef]
40. Bolaji, A.L.; Babatunde, B.S.; Shola, P.B. Adaptation of Binary Pigeon-Inspired Algorithm for Solving Multidimensional Knapsack Problem. In *Soft Computing: Theories and Applications*; Advances in Intelligent Systems and Computing; Springer: Singapore, 2018.
41. Li, G.; Chou, W. Path planning for mobile robot using self-adaptive learning particle swarm optimization. *Sci. China Inf. Sci.* **2018**, *61*, 052204. [CrossRef]
42. Nikolos, I.K.; Brintaki, A.N. Coordinated UAV Path Planning Using Differential Evolution. *Oper. Res.* **2005**, *5*, 487–502.