aerospace

MDPI

Article

Global Optimization of UAV Area Coverage Path Planning Based on Good Point Set and Genetic Algorithm

Jinbiao Yuan [1], Zhenbao Liu [1,2,*], Yeda Lian [1,*], Lulu Chen [1], Qiang An [3], Lina Wang [1] and Bodi Ma [1]

[1] School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China; yuanjinbiao0414@126.com (J.Y.); chenlulu@mail.nwpu.edu.cn (L.C.); lena@mail.edu.cn (L.W.); mabodi0112@126.com (B.M.)
[2] Research & Development Institute in Shenzhen, School of Civil Aviation, Northwestern Polytechnical University, Shenzhen 518057, China
[3] Research Institute of Aero-Engine, Beihang University, Beijing 100190, China; anqiang@buaa.edu.cn
[*] Correspondence: liuzhenbao@nwpu.edu.cn (Z.L.); RunnerLian01@163.com (Y.L.)

Abstract: When performing area coverage tasks in some special scenarios, fixed-wing aircraft conventionally adopt the scan-type of path planning, where the distance between two adjacent tracks is usually less than the minimum turning radius of the aircraft. This results in increased energy consumption during turning between adjacent tracks, which means a reduced task execution efficiency. To address this problem, the current paper proposes an area coverage path planning method for a fixed-wing unmanned aerial vehicle (UAV) based on an improved genetic algorithm. The algorithm improves the primary population generation of the traditional genetic algorithm, with the help of better crossover operator and mutation operator for the genetic operation. More specifically, the good point set algorithm (GPSA) is first used to generate a primary population that has a more uniform distribution than that of the random algorithm. Then, the heuristic crossover operator and the random interval inverse mutation operator are employed to reduce the risk of local optimization. The proposed algorithm is verified in tasks with different numbers of paths. A comparison with the conventional genetic algorithm (GA) shows that our algorithm can converge to a better solution.

Keywords: UAV; area coverage; GA; path planning; GPSA

check for updates

Citation: Yuan, J.; Liu, Z.; Lian, Y.; Chen, L.; An, Q.; Wang, L.; Ma, B. Global Optimization of UAV Area Coverage Path Planning Based on Good Point Set and Genetic Algorithm. *Aerospace* **2022**, *9*, 86. https://doi.org/10.3390/aerospace9020086

Academic Editor: Kamil Krasuski

Received: 24 December 2021
Accepted: 1 February 2022
Published: 7 February 2022

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have been widely used in mapping, photography, search and rescue, strike, reconnaissance, and mountain, urban, highway, and oil/gas inspections. As part of path planning, coverage path planning (CPP) [1–3] of UAV is also very important in mapping, searching, and patrol inspection [4–7]. Among commonly used UAVs, rotor UAVs are flexible, can hover and vertically lift, and have low requirements on the take-off site, but at the same time, suffer from small load, low cruise speed, and short endurance time. On the other hand, fixed-wing UAVs are less flexible than rotorcraft, but they have heavy load, fast cruise speed, and long endurance. Therefore, the use of rotorcraft has advantages in missions featuring small areas, large terrain fluctuations, fixed-point lighting, fixed-point shouting, and short-range high-precision tracking shooting, whereas for large-area and long-distance area coverage, patrol inspection, mapping, and search, fixed-wing UAVs are obviously more suitable.

Fixed-wing UAVs in area CPP applications usually plan a scan-type path (a group of parallel lines covering a certain area that are executed in turns, and no crossing of lines is allowed) based on the actual need. While such demand can be met most of the time, in some special applications such as digital orthophoto map (DOM) generation and tilt photography model generation, the lateral distance between two adjacent planned paths is usually less than the UAV's turning radius. This increases the distance traveled by the fixed-wing UAV, leading to additional energy consumption and thereby affecting the UAV's flight time. In path planning, when the area is a convex polygon, conventional algorithms can quickly and accurately calculate the width of the search area and find the corresponding supporting parallel line. When the UAV flies along the parallel line, fewer turns lead to a shorter path and lower energy consumption [8]. By taking the turning path length between two paths as the distance between them, and setting the take-off point as the landing point, the area coverage problem can be transformed into a traveling salesman problem (TSP) [9]. In this paper, we will use the genetic algorithm to solve the covering path problem with its genetic evolution operation.

Genetic algorithm (GA) is a population search algorithm [10] with strong global search ability, especially in complex optimization problems, which makes GA a popular choice in path planning, task allocation, etc. At present, several studies focus on GA path planning including track coding [11], design and improvement of GA operator, and generation of population, where researchers hope to solve the problems associated with precocity and convergence speed in GA. Liu et al. [12] propose a mobile robot path planning method based on the adaptive genetic algorithm. This method uses special genetic operators and adaptively adjusts parameters to plan the path. Feasibility and effectiveness of the algorithm are also experimentally verified in this study. Qu et al. [13] report a coevolutionary improved genetic algorithm (CIGA) for global path planning of multiple mobile robots, which effectively solves the local optimization problem and accelerates convergence. Lee et al. [14] put forward an effective initialization method for robot path planning based on genetic algorithm. Tuncer et al. [15] propose a new mutation operator and verify its performance by applying it in the dynamic environment. Filho et al. [16] propose a new heuristic algorithm for combinatorial optimization. This algorithm is based on genetic algorithm and betting theory, which applies probabilistic recombination to a solution given as input and returns a new solution that is much more optimized. Holst [17] demonstrates that the genetic algorithm is a reliable and flexible approach in optimization. Li et al. [18] propose a station coding based genetic algorithm to solve the satellite range scheduling problem with priority constraints. This algorithm adopts a new chromosome encoding method that arranges tasks according to the ground station ID, which reduces the complexity in conflict checking and resolving and helps find the optimal resolution. Liu et al. [19] extend the application of the Linkage Learning Genetic Algorithm (LLGA) to other flight conditions and then simulates a complete flight mission with different gains and weather condition configurations. The simulation results show that the engine performance has been greatly improved after optimization by LLGA in the transient state and the high-altitude conditions.

The good point set algorithm (GPSA) can construct uniformly distributed solution sets in high-dimensional space [20–22]. In this paper, the area coverage problem is converted into a TSP problem, and the polygons are divided into parallel paths with certain intervals. These parallel paths are numbered (gene coding) based on a genetic algorithm. The good point set algorithm is then used to construct the primary population with the principle of no repeated access to the same node. The crossover and mutation operations of the genetic algorithm are also improved. Since the initial population distribution obtained by GPSA is more uniform than that obtained by the random algorithm, a better solution can often be acquired in the evolutionary operation.

The main contributions of this paper are as follows: (1) use of the good point set algorithm to generate high-quality primary population for GA, (2) achievement of a better solution compared to conventional GA, with the adoption of heuristic crossover operator and random interval reverse mutation to improve the crossover and mutation operators for GA, and (3) improvement in the covered path length in the same area compared to the traditional scan-type algorithm.

The remainder of this paper is organized as follows. Section 2 introduces existing work related to coverage path planning. Section 3 presents the kinematic model of UAV. Conversion from CPP problem to TSP problem is described in Section 4, followed by the design of the improved genetic algorithm in Section 5. Section 6 shows the results from the theoretical experiment and actual flight experiment. Conclusions and future work are given in Section 7.

2. Related Work

The objective of CPP is to enable the intelligent agent to traverse all positions in the coverage area according to some required overlap rate within the detection range of airborne sensors. At present, CPP studies mainly focus on ground robots, whereas the research on UAV path coverage is insufficient. The UAV coverage path planning studied in this paper is based on convex polygons; non-convex polygons can also be transformed into convex polygons through segmentation operation or connecting the intersections of both sides of concave edges.

An energy-aware spiral CPP algorithm for UAV photogrammetry is proposed in [23], which considers the camera sensor and flight altitude when executing the overlap required to ensure mission success. The algorithm uses the energy model to set optimal speeds for the different straight-line segments of the path, so as to reduce energy consumption. The energy model is also improved to predict the total energy of the path. Real flights performed by a quadrotor demonstrate the effectiveness of the E-Spiral over E-BF in two areas, with an improvement of 9% in time and 7.7% in energy, respectively. The estimation precision of time and energy is also improved by13.24% and 13.41%, respectively. However, this quadrotor-based study does not consider the situation of fixed-wing UAV and its kinematic performance, Moreover, for a polygon with a small angle but a long edge of the angle, the coverage of the included angle cannot be achieved efficiently by this method.

Reference [24] proposes a hierarchical heuristic search algorithm for UAV coverage planning based on Gaussian mixture model. The algorithm prioritizes the searching subregions by using the pattern superiority ratio heuristic algorithm and Gaussian mixture model. It can search for effective paths through the parameter space at different levels of resolution. Experiments show that the performance of the new algorithm is significantly better than the existing algorithms (Bourgault's algorithm and LHC-GW-CONV algorithm). This algorithm can be applied to decentralized and local areas in search/rescue scenes. Its core purpose is to search targets quickly and effectively, which is different from the area coverage scene studied in this paper. Moreover, when such a trajectory is executed by a fixed-wing UAV, there will be many redundant paths due to the influence of the minimum turning radius.

Reference [25] systematically studies the TSP-CPP problem. The authors provide a mixed integer programming formula for this new problem, and then introduce a CPP method covering a single convex polygon region. Through comprehensive theoretical analysis and simulation research, they show the optimality and effectiveness of the proposed method. In the study, coverage of multiple decentralized areas is conducted by a single UAV. The decentralized regions are abstracted as TSP problems and solved by integer programming to obtain the order of access. The benchmark CPP methods are then used to plan the coverage path of the single convex polygon area. Similar to [23], the kinematic constraints of fixed-wing UAV in a single convex polygon are not considered here.

A method for generating the path of agricultural fixed-wing UAV covering arbitrary polygon area is proposed in [26]. By considering the influence of wind on the ground track of UAV, this method minimizes time consumption. Assuming that the path must be a set of parallel lines and cover the field with strips, this algorithm determines the best angle of these strips, as well as the best order and direction of access. Numerical simulations show the dependence of the obtained flight path on the wind field geometry. This problem is similar to the problem studied in our present paper. However, only a rotor UAV is used in [26]. In addition, when solving the parallel coverage path of the TSP problem, the distance between the two points is calculated only according to the actual coordinates of the two ends of the parallel line segment. This works well for rotor UAVs due to their superior kinematic characteristics, but for fixed-wing UAVs, especially when turning from one path to adjacent paths, due to the influence of their turning radius, the actual distance between adjacent paths will be greater than the coordinate straight-line distance of their endpoint. In the present study, the turning distance of the fixed-wing UAV will be thoroughly considered, so that the real-world situation can be better represented.

In order to provide long-term communication coverage for ground mobile users, a fully distributed control solution is provided in [27], which can be used to navigate a group of UAVs flying around the target area as a mobile base station (BSS). Specifically, each UAV is controlled in a distributed manner through a framework based on decentralized deep reinforcement learning (DRL). The algorithm applies the DRL theory to design the state, observation, action space, and reward. After a large number of simulation experiments, an appropriate set of super parameters is found, including the size of the empirical playback buffer, the number of neural units of two fully connected hidden layers of actors, critics, and their target network, and the discount factor to remember the future reward. The algorithm can maximize the time average coverage fraction of all UAVs in the mission, maximize the geographical fairness of all considered points of interest (POI), and minimize the total energy consumption, while keeping them connected and not flying out of the regional boundary. This is apparently a multi-UAV coverage problem, different from the present study which is based on a single UAV. However, the ideas of regional coverage optimization from [27] are enlightening.

3. Kinematic Model of UAV

In area coverage path planning, for given dynamics constraints of the UAV, fewer turns generally mean a shorter path. In this paper, a fixed-wing UAV with vertical take-off and landing (VTOL) is used as the carrier to study the problem of regional coverage path planning. Considering the influence of terrain change, in order to simplify the problem, the following reasonable assumptions are made.

(1) The area coverage is set to meet the zoom range requirements of the aerial photography equipment for all encountered to terrain fluctuations, to ensure that the UAV can perform tasks at the same altitude.

(2) The UAV is within the error range during turning and straight-line following, and situations such as gust and sideslip are not considered.

(3) Each area covering the path planning task is a convex polygon (When the shape of the region is a concave polygon, the concave parts are connected to form a convex polygon for processing).

The VTOL UAV is constrained by battery power and cannot switch endlessly between rotor and fixed-wing modes. Usually, the rotor mode is used for take-off and landing, whereas the fixed-wing mode is employed in cruise to save energy. Therefore, path planning of a VTOL UAV should meet its kinematic constraints as much as possible. According to the assumptions above, the UAV coverage path planning can be studied on a two-dimensional plane. When flight speed and load of the UAV are known, only the minimum turning radius constraint needs to be considered. The Kinematic parameters and variable description of the UAV are shown in Figure 1.

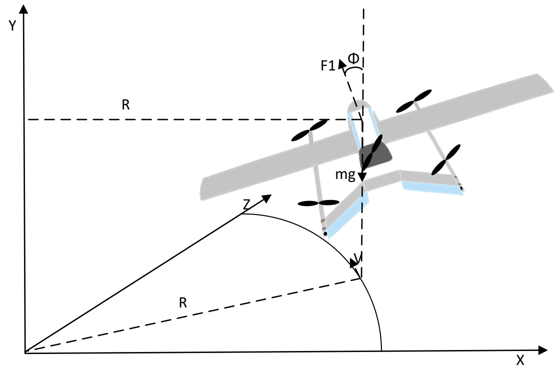


Figure 1. Kinematic parameters and variable description of the UAV.

According to Figure 1, the following relationship can be obtained.

$$\left.\begin{array}{c} F1 * \cos(\Phi) = mg \\ \frac{m*V^2}{R} = F1 * sin(\Phi) \end{array}\right\} \tag{1}$$

where $F1$ is the lift of UAV, m is the mass of the aircraft, mg is the gravitational force exerted on the UAV (including airborne equipment), Φ the rolling angle of UAV, R the minimum turning radius of UAV, V the instantaneous speed of UAV, $V \in (V_{min}, V_{max})$, and g the gravitational acceleration. We can also write

$$\mathrm{R} = \frac{{V_{min}}^2}{g * tan(\Phi)} \tag{2}$$

To obtain an optimal path, the minimum turning radius should be met during turning.

4. Problem Formulation

A. TSP problem conversion

The area that a single fixed-wing UAV can fly under energy constraints is limited. Path planning normal to the width direction of the convex polygon is the optimal strategy [1], as shown in Figure 2.

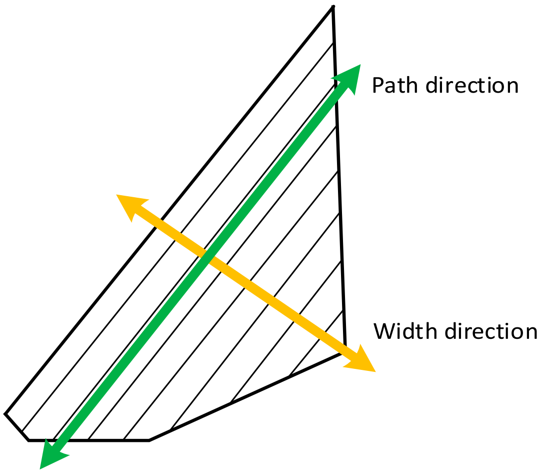


Figure 2. Optimal path division.

In Figure 2, the green thick line indicates the direction of the path, and the thick yellow line indicates the width direction. However, if the path is executed in sequence, e.g., one by one from left to right, the UAV needs to fly longer when turning between paths, resulting in excess energy consumption, because the distance between two adjacent paths is less than the minimum turning radius. Therefore, three situations can be considered when turning between adjacent tracks, as shown in Figure 3.

Three area coverage strategies are shown in Figure 3, where the dotted line represents the path to be flown by the UAV in the mission area, and the trapezoidal region represents the area covered by each track. The yellow line is the path that the UAV needs to fly outside the mission area. It is obvious that (b) is the best turning strategy. Strategy (c) could be the second best in a certain range. Strategy (a) seems the worst. However, beyond a certain distance range, (a) could be better than (c).

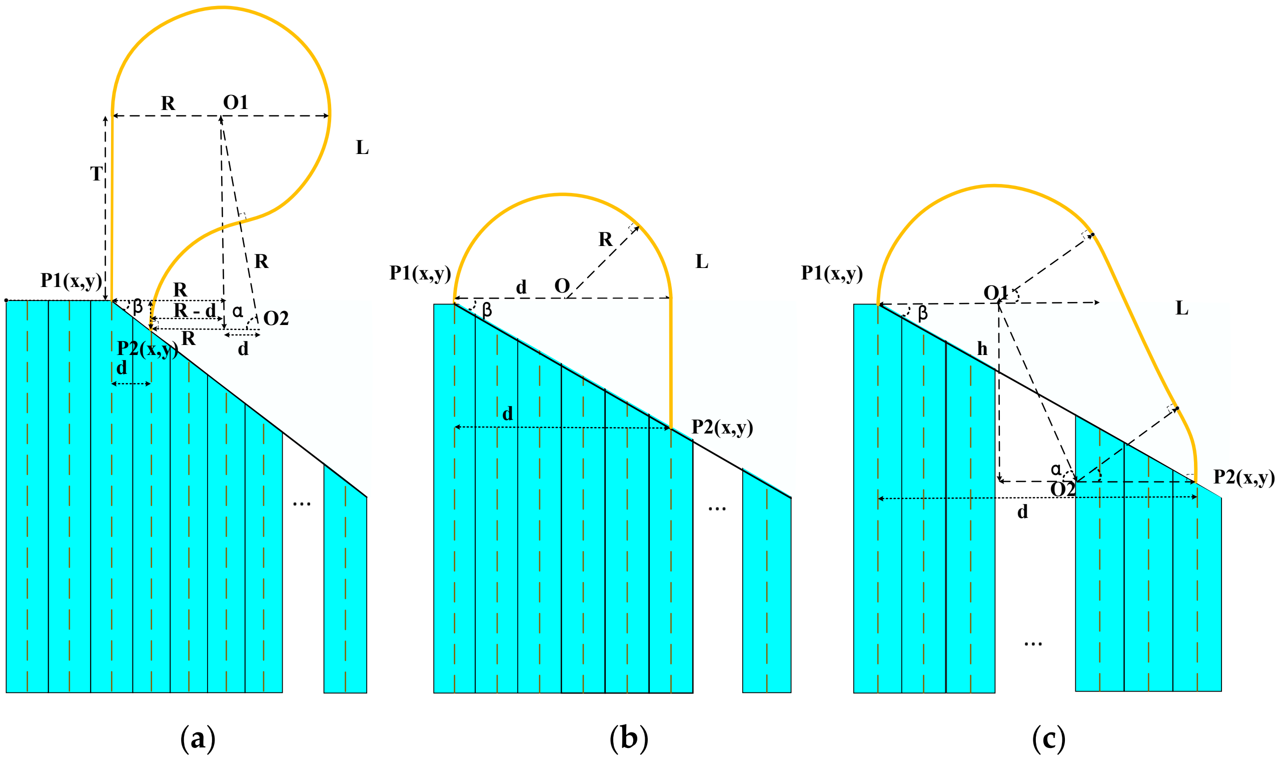


Figure 3. Different scenarios of turning: (**a**) span less than turning radius, (**b**) span equal to turning radius, and (**c**) span greater than turning radius.

It can be seen from Figures 1 and 2 that the regional coverage path planning task can be fulfilled as long as the aircraft can complete these paths, and that different path sequences lead to different path lengths. Another important constraint is that the aircraft would have to take off and land at the same point in practical operation. Therefore, this problem can be transformed into a tsp [28–30]. The specific treatment is as follows.

(1) number each path in sequence $N = 1, 2, 3 \cdots$.

(2) both ends of each path can be used as either entry point or departure point of that path; whether entry or departure will be determined by the starting point.

(3) Calculate the actual distance from one path to another. The distance L is calculated with reference to the annotations in Figure 2, from the formula below.

$$L = \begin{cases} T + \pi * R + 2\alpha * R, \;\; d < R \\ \pi * R + d * \cos(\beta), \;\; d = R \\ \pi * R + \sqrt{(d-2R)^2 + h^2}, \; d > R \end{cases} \tag{3}$$

where R is the minimum turning radius, d is the span, $\alpha = \arccos\left(\frac{d}{2R}\right)$, $\beta = \arctan(abs(P1.y - P2.y)/abs(P1.x - P2.x))$, $T = R * \sin(\alpha) - d * \tan(\beta)$, $h = abs(P1.y - P2.y)$.

B. Waypoint generation

Since the turning path tracking of the considered VTOL UAV adopts L1 [31–33] algorithm we need, therefore, to calculate a reasonable waypoint to ensure that the UAV can fly in the optimal path when turning. Each waypoint reached by the UAV has an acceptance radius; when considering the optimal path, the acceptance radius is equal to the turning radius.

The three cases a, b and c in Figure 4 correspond to a, b and c in Figure 3. The pseudo code of waypoint generation is as follows (Algorithms 1 and 2):

Algorithm 1. waypoint_generate.

```
Input : Route Sequence RVector, Sequence size N
Init : newVecotr
for epoch = 1,2...N − 1 do
    line1 ⟵ RVecor[epoch]
    line2 ⟵ RVecor[epoch + 1]
    P1′(lon, lat) ⟵ Ned2Geo(line1.P1)
    P1(lon, lat) ⟵ Ned2Geo(line1.P2)
    P2′(lon, lat) ⟵ Ned2Geo(line2.P1)
    P2(lon, lat) ⟵ Ned2Geo(line2.P2)
    newVector.append{P1′, P1}
    tempVector ⟵ sequential_line_waypoint_generate(P1, P2, P1′, P2′, R)
    newVector.append(tempVector)
    newVector.append{P2′, P2}
end for
return newVecotr
```

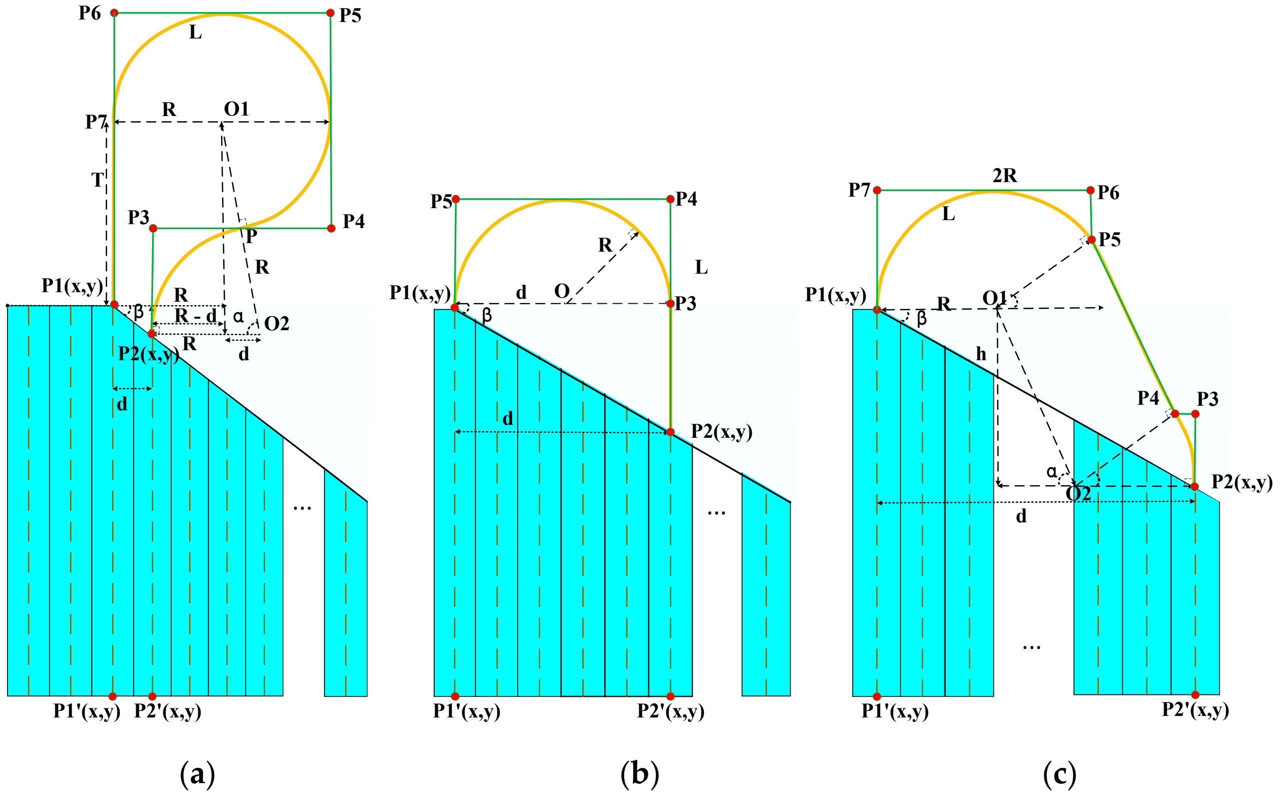


Figure 4. Waypoint generation under known trajectory: (**a**) span less than turning radius, (**b**) span equal to turning radius, and (**c**) span greater than turning radius.

Ned2Geo converts the Ned (North East Down) coordinate system to the geographic coordinate System, Geo. Geo2Ned does just the opposite.

Algorithm 2. sequential_line_waypoint_generate.

```
Input: Geo: P1(lon, lat), P2(lon, lat), P1′(lon, lat), P2′(lon, lat), minimum turning radius R
Ned: $P1(x,y) \longleftarrow Geo2Ned(P1(lon,lat))$, $P2(x,y) \longleftarrow Geo2Ned(P2(lon,lat))$
Init: ReultVector
$\beta \longleftarrow atan\left(\frac{abs(P2.y-P1.y)}{abs(P2.x-P1.x)}\right)$
$d \longleftarrow distance(P1(x,y), P2(x,y)) * \cos(\beta)$
mark $\longleftarrow$ compare(d, 2*R) where mark = {0, 1, 2} and corresponds to three cases a, b and c
switch(mark)
{
    case 0:
        azimuth $\longleftarrow$ P1′(lon, lat).azimuthto(P1(lon, lat))
        vazimuth $\longleftarrow$ LineDirectionAzimuth $+ \pi*0.5$
        $\alpha \longleftarrow acos(d/2R)$
        P3 $\longleftarrow$ P2(lon, lat).atDistanceAndAzimuth($R*\cos(\alpha)$, azimuth)
        P4 $\longleftarrow$ P3(lon, lat).atDistanceAndAzimuth($2R - d$, $-$vazimuth)
        P5 $\longleftarrow$ P4(lon, lat).atDistanceAndAzimuth($2R - d$, $-$vazimuth)
        P6 $\longleftarrow$ P5(lon, lat).atDistanceAndAzimuth(2R, $-$vazimuth)
        P7 $\longleftarrow$ P6(lon, lat).atDistanceAndAzimuth(R, $-$azimuth)
        ReultVector.append{p3, p4, p5, p6, p7}
            break;
    case 1:
        azimuth $\longleftarrow$ P1′(lon, lat).azimuthto(P1(lon, lat))
        vazimuth $\longleftarrow$ LineDirectionAzimuth $+ \pi*0.5$
        P3 $\longleftarrow$ P2(lon, lat).atDistanceAndAzimuth($d * \tan(\beta)$, azimuth)
        P4 $\longleftarrow$ P3(lon, lat).atDistanceAndAzimuth(R, azimuth)
        P5 $\longleftarrow$ P4(lon, lat).atDistanceAndAzimuth(2R, $-$vazimuth)
        ReultVector.append{p3, p4, p5}
            break;
    case 2:
        azimuth $\longleftarrow$ P1′(lon, lat).azimuthto(P1(lon, lat))
        vazimuth $\longleftarrow$ LineDirectionAzimuth $+ \pi*0.5$
        $h \longleftarrow abs(P1.y - P2.y)$
        $\alpha \longleftarrow atan\left(\frac{h}{d-2R}\right)$
        P3 $\longleftarrow$ P2(lon, lat).atDistanceAndAzimuth($R*\cos(\alpha)$, azimuth)
        P4 $\longleftarrow$ P3(lon, lat).atDistanceAndAzimuth($R - R*\sin(\alpha)$, $-$vazimuth)
        P5 $\longleftarrow$ P4(lon, lat).atDistanceAndAzimuth($h/\sin(\alpha)$, $-$vazimuth $+ \alpha$)
        P6 $\longleftarrow$ P5(lon, lat).atDistanceAndAzimuth($R - R*\cos(\alpha)$, azimuth)
        P7 $\longleftarrow$ P6(lon, lat).atDistanceAndAzimuth($R + R*\sin(\alpha)$, $-$vazimuth)
        ReultVector.append{p3, p4, p5, p6, p7}
            break;
    default:
            break;
}
return ReultVector
```

5. Improved Genetic Algorithm

Genetic algorithm (GA) belongs to the family of bionic algorithms. It is a random search algorithm based on biological natural selection and genetic mechanisms, with the advantages of parallel search and global optimization [14].

5.1. Encoding of Individual Genes

After numbering the area coverage paths, it is necessary to access all path numbers without repetition (except for the take-off and landing path numbers). Here, the path numbers are regarded as genes, and the chromosome length of each individual of the population is fixed. Since in an actual operation, the take-off and landing points are always chosen in advance, the starting number of each chromosome is the same.

5.2. Initial Population Generation

Generation of initial population in the conventional genetic algorithm usually adopts the principle of random generation, which is susceptible to falling into local optimization in the iterative process. According to the definition and properties of the good point set [34–36], when the integral of the function on the unit cube G_s in the s-dimensional Euclidean space is approximated by any weighted sum of the function values of any given n points, the resultant error of taking n good points reaches minimum, with a point distribution that is more uniform than those obtained by other means. Similarly, mapping the good points in G_s space to the solution space results in a solution set that is more evenly distributed and closer to the solution space. In other words, there is a higher chance to obtain the optimal solution. Generally, construction of the good point set is independent of the spatial dimension, overcoming the shortcomings of the orthogonal table design method. It also provides a very good means for high-dimensional approximation. Therefore, in order to improve the solution accuracy, this paper uses the good point set algorithm to construct the initial population, through which a more evenly distributed solution set can be realized in the solution space. In these populations, the evolutionary operation can evolve to the optimal solution with a greater probability than the random population. The steps of constructing the initial population from the good point set are listed below.

(1) Let G_s be the unit cube of an S-dimensional Euclidean space. Let $m \in G_s$, $m = (m_1, m_2, \ldots, m_s), 0 \le m_i \le 1, i = 1, 2, \ldots, s$;

(2) Let $P_n(k) = \{\left(x_1^{(n)}(k), x_2^{(n)}(k), \ldots, x_s^{(n)}(k)\right)$ be a point set in $G_s, 0 \le k \le n, 0 \le x_i^{(n)}(k) \le 1, 1 \le i \le s\}$, where n denotes the size of the point set;

(3) Let $r \in G_s$, shaped as $P_n(k) = \left\{\left(r_1^{(n)}(k), r_2^{(n)}(k), \ldots, r_s^{(n)}(k)\right), 0 \le k \le n\right\}$, with its deviation $\varphi(n)$ satisfying $\varnothing(n) = C(r,\varepsilon)n^{-1+\varepsilon}$, where $C(r,\varepsilon)$ is a constant only related to $r, \varepsilon(\varepsilon > 0)$. Then, $P_n(k)$ is considered as a good set of points and r is called a good point;

(4) Taking $r_k = \left\{2cos\frac{2\pi k}{p}\right\}, 1 \le k \le s$, where p is the smallest prime number satisfying $\frac{p-3}{2} \ge s$, r is a good point. If $r_k = \left\{e^k\right\}$, $1 \le k \le s$, then r is also a good point.

A group of initial populations with uniform distribution can be constructed using the algorithm above. In the solving process, depending on the different population sizes, the calculation results are all floating-point values. In the gene coding, the numbering of paths requires integers. Therefore, the good point set should be rounded, after which duplicate numbered paths will be generated within each individual in the population. Then these duplicates need to be replaced with paths whose sequence numbers are close to those integers that are not in the current numbering list.

5.3. Fitness Function

Genetic algorithm uses the fitness value of individuals in the population to search. Selection of the fitness function directly affects convergence speed of the algorithm and whether the optimal solution can be found. The focus of this study is essentially a distance minimization problem, with a fitness function of

$$fit(f) = -\sum_{i=0}^{N}(s_i - s_{i+1}) \tag{4}$$

where f denotes each individual, and N denotes the number of genes in individual chromosomes, s_i denotes the ith gene. The fitness function can be used to measure the advantages and disadvantages of individuals, so that the algorithm can carry out subsequent genetic operation. The distance between different individuals is calculated from Equation (3).

5.4. Genetic Operation

Genetic operation includes three parts: selection, crossover, and mutation. After the initial population is generated, the optimal solution can be sought through iterative genetic operation. In genetic algorithm, the crossover operator is the main operator because of its global search ability, whereas the mutation operator is regarded as the auxiliary operator due to its local search ability. Thus, genetic algorithm has fairly balanced capabilities of both global and local search, as a result of cooperation and competition between crossover and mutation. The so-called cooperation means that when the population is trapped in a hyperplane in the search space and fails to escape by only the crossover operation, the mutation operation can help it jump out of the current limit and make the iteration converge to a better solution. Competition means that when the desired chromosomes have been obtained through cross operation, the mutation operation may damage these chromosomes and degrade individual fitness. The settings of genetic operation in this study are as follows.

5.4.1. Heuristic Crossover Operator

Treat the two gene positions with the largest distance in the chromosome with a large fitness as the initial crossover position, and execute cross operation on them, with the goal of obtaining individuals with better fitness than their parents. As shown in Figure 5, the specific steps are listed below.

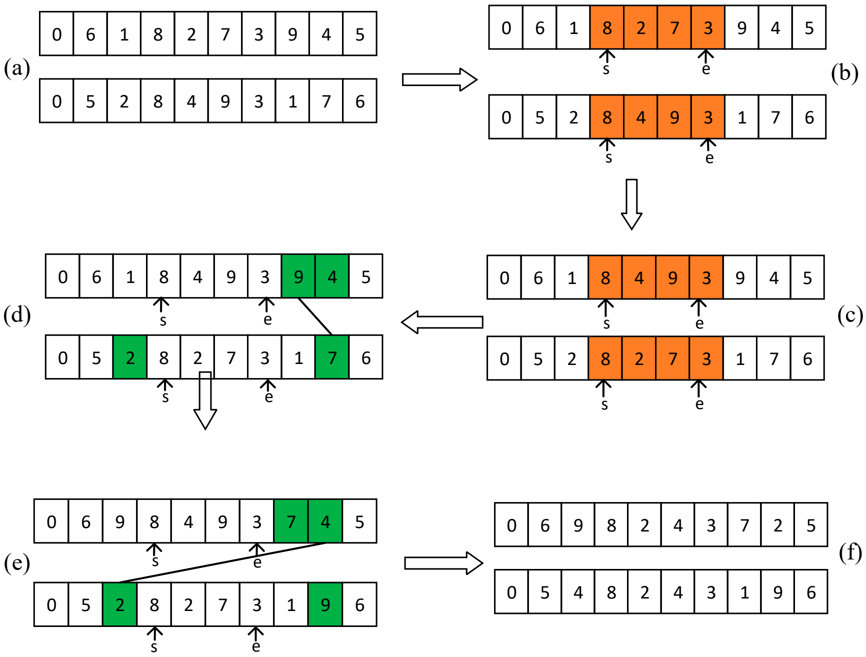


Figure 5. Heuristic crossover operation: (**a**) two adjacent individuals in a population, (**b**) determine the starting point and ending point of crossover operation in individual gene sequence, (**c**) exchange the genes from *s* to *e* between the two chromosomes, (**d**,**e**) the duplicate genes in each chromosome are collected from both sides of *s* and *e*, (**f**) two individuals after cross operation.

(1) Disrupt the existing population order and select two adjacent individuals with the set crossover probability, as shown in Figure 5a;
(2) Select the individual with large fitness value from step (1), and calculate the distance between its two adjacent genes. Select the latter of the two genes with the largest distance as the intersection starting point. As shown in Figure 5b, *s* is the intersection

starting position, and the corresponding intersection starting position of another chromosome is also s. The intersection ending point can be randomly obtained on the other side of the starting point. As shown in Figure 5b, e is the intersection end position;

(3) Exchange the genes from s to e between the two chromosomes, as shown in Figure 5c;

(4) The duplicate genes in each chromosome are collected from both sides of s and e. Calculate chromosome fitness values. Individuals with a large fitness value are preferentially replaced with duplicate genes, as shown in Figure 5d;

(5) Repeat step (4) until the repeated gene replacement is completed. Now, one of the two genes has a larger fitness value whereas the other has a smaller one. Diversity of the population can be maintained, as shown in Figure 5f.

5.4.2. Random Interval Reverse Mutation

The purpose of introducing mutation in genetic algorithm is twofold. First, to endow the genetic algorithm with the ability of local random search. When the algorithm has approached the neighborhood of the optimal solution through the crossover operator, local random search performed by the mutation operator can accelerate convergence to the optimal solution. Of course, the mutation probability in this case should be set to a small value, otherwise the trend of approaching the optimal solution could be hindered by mutation. Another important function of mutation is to help genetic algorithm maintain population diversity, thereby preventing premature convergence. As shown in Figure 6, the mutation operation follows the steps described below.

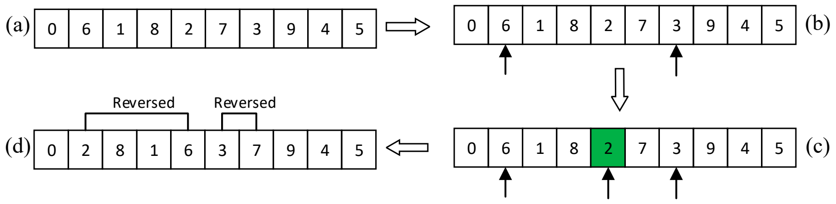


Figure 6. Mutation operation: (**a**) randomly extract an individual from the population with a preset mutation probability, (**b**) Randomly obtain the start and end position indices for the mutation operation, (**c**) calculate the middle position index between the start and end position indices, (**d**) Reverse the order of the numbers in two intervals.

(1) Randomly extract an individual from the population with a preset mutation probability, as shown in Figure 6a;

(2) Randomly obtain the start and end position indices for the mutation operation, as illustrated by genes 6 and gene 3 in Figure 6b;

(3) Calculate the middle position index between the start and end position indices (green square in Figure 6c;

(4) Reverse the order of the numbers in two intervals: (a) from the start position index to the middle position index, and (b) from middle to end, in order to obtain the variant individuals, as shown in Figure 6d.

5.4.3. Tournament Selection

Subsequent to crossover and mutation operations, the current population is a new population, in which the individuals possess a richer diversity than that before the genetic operation. However, since only those population individuals with a large fitness will be selected, the differences between chromosomes tend to be gradually homogenized after several generations. Therefore, the tournament method is adopted here. That is, several individuals are sampled with replacement from the population with a set probability; repeating this operation for many times will result in the scale of the initial population. The specific steps are as follows.

(1) Determine the number of individuals K for each selection and the number of selections C;
(2) Randomly select M individuals with replacement from the population, within which K individuals with the highest fitness are further selected to enter the offspring population;
(3) Repeat step (2) multiple times until the C is reached. In other words, the population size should reach the size of the primary population to form a new generation of population.

5.5. Algorithm Description

The overall flow of the algorithm proposed in this study is shown in Figure 7.

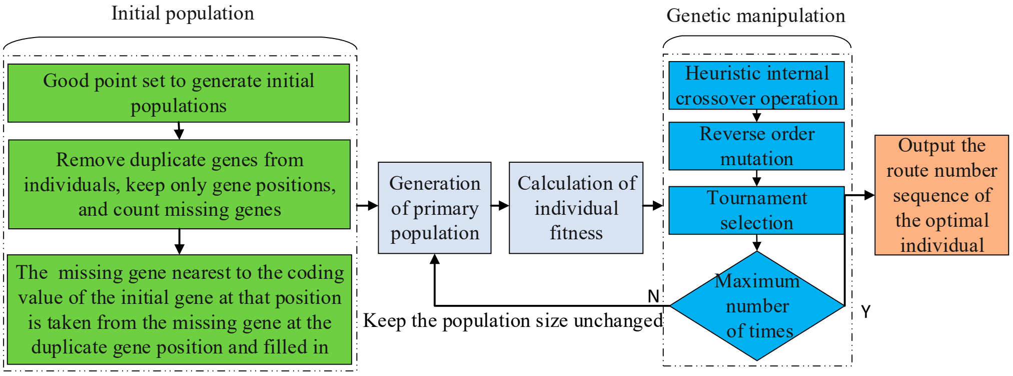

Initial population
Good point set to generate initial populations
Remove duplicate genes from individuals, keep only gene positions, and count missing genes
The missing gene nearest to the coding value of the initial gene at that position is taken from the missing gene at the duplicate gene position and filled in
Generation of primary population
Calculation of individual fitness
Genetic manipulation
Heuristic internal crossover operation
Reverse order mutation
Tournament selection
Maximum number of times
N
Y
Keep the population size unchanged
Output the route number sequence of the optimal individual

Figure 7. Genetic algorithm flow chart.

Firstly, a uniformly distributed population with P individuals is obtained by the good point set algorithm. Then, a new population is created through crossover, mutation, and selection from the genetic algorithm. After N iterations, a population with a high average fitness value can be reached. Individuals with the highest fitness are subsequently selected, which, in other words, give the optimal track sequence number. Finally, by selecting an endpoint from the starting number as the take-off point, the complete track planning can be carried out based the sequence of track numbers. As shown in Figure 8, the optimal track numbering sequence is 0, 6, 1, 7, 2, 8, 3, 9, 4, 5, 0. Taking path 0 as the starting point, complete planning can be performed accordingly. It is noted that paths 4 and 5 are adjacent paths, and their separation is shorter than the minimum turning radius of the UAV. Therefore, when the UAV enters 5 from 4, planning is conducted according to Figure 3a; the other two scenarios should follow Figure 3b,c.

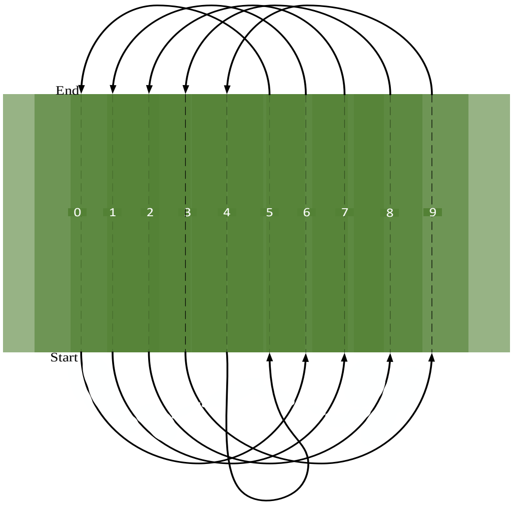


Figure 8. Schematic diagram of regional coverage path planning.

6. Experiment and Analysis

6.1. Algorithm Test

In algorithm verification, both GPSA-GA and GA are tested 20 times in coverage areas of 20, 50, 80, and 100 paths, respectively. The parameters to be compared mainly include the number of iterations at convergence (NIC), the optimal path length (OPL), and the execution time (ET). Calculation of the path length ignores the path length in the coverage area (because each algorithm passes through the same path during internal execution of each path). Only the path length affected by the turning radius between paths needs to be considered. Relevant algorithm parameters are listed in Table 1.

Table 1. List of main parameters of the algorithm.

Algorithm	Parameters	Value	Note
GPSA − GA	Number of iterations	10,000	
	Population size	60	
	Chromosome length	N	N = {20,50,80,100}
	Crossover probability	0.8	
	Mutation probability	0.05	
	Percentage of selection	60%	
GA	Number of iterations	10,000	
	Population size	60	
	Chromosome length	N	N = {20,50,80,100}
	Crossover probability	0.8	
	Mutation probability	0.25	
	Percentage of selection	60%	
Global parameters	R	180 m	
	Path spacing	60 m	

Distance calculation in the algorithm test is carried out according to Equation (3), where only the distance passed during turning is considered. Converging processes to the optimal values are shown in Figure 9.

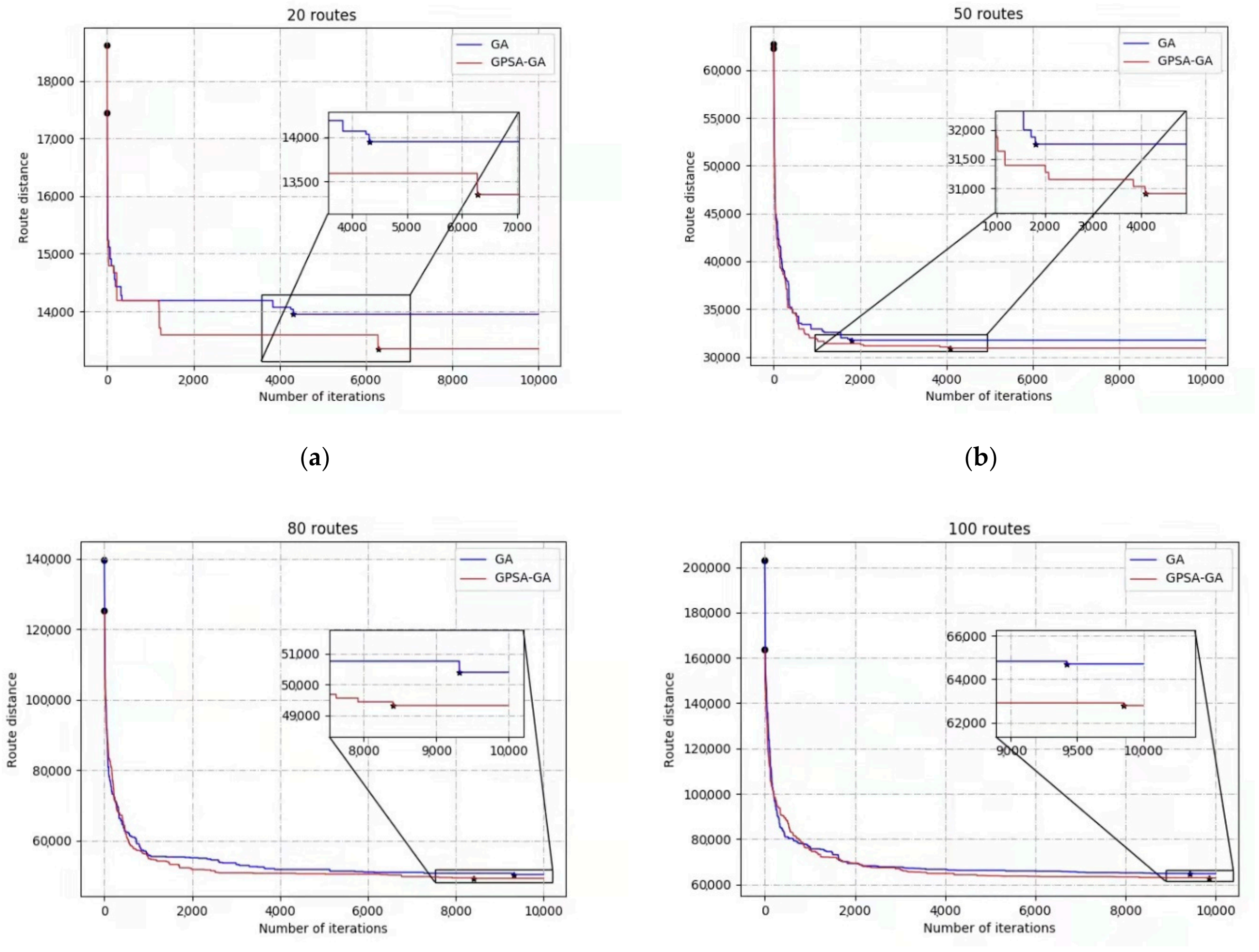
In Figure 9, "*" denotes the first value at which the algorithm begins to converge, and "•" the optimal individual of the initial population. As can be seen from Figure 9a, compared with GA, GPSA-GA obtains better solutions in all four scenarios with higher-quality offspring.

Detailed comparison in terms of OPL, NIC, and ET are provided in Table 2.

Table 2. Average comparison.

Algorithm	Indicator	20 Paths	50 Paths	80 Paths	100 Paths
GPSA − GA	OPL (m)	13,574.89	31,805.5	49,949.34	62,991.54
	NIC	4889.9	5099.5	8111.25	8901.75
	ET (s)	11.44	28.98	79.31	113.65
GA	OPL (m)	13,641.55	32,052.45	50,565.09	64,112.09
	NIC	5027.95	4970.45	7738.7	9135.05
	ET (s)	11.17	26.25	65.8	100.68

For 20, 50, 80, and 100 paths, OPLs obtained by GPSA-GA are all better than those by GA. That is, GPSA-GA achieves larger improvements relative to the corresponding scan paths (shown in Table 3). Moreover, such improvement is more significant for a higher numbers of paths. The initial generation population created by the good point set algorithm has a more uniform distribution than the random population, thereby having a higher chance to find the optimal solution. However, constructing the initial population in the solution space will increase the convergence time of GPSA-GA. In the subsequent calculation, the needed number of iterations basically depends on the genetic mutation operation but has little to do with the initial solution set.



(**a**) (**b**)

(**c**) (**d**)

Figure 9. Optimal value convergence trend. (**a**) 20 paths, (**b**) 50 paths, (**c**) 80 paths, (**d**) 100 paths.

Table 3. Scan-type path length.

Number of paths	20	50	80	100
Scan-type path length (m)	31,312	80,752	130,192	163,152

6.2. Path Planning

The path planning effects of different area shapes are shown in the following figure.

In Figure 10a,b are the path planning for convex polygon regions, Figure 10c,d are path planning for concave polygons, Figure 10e,f are path planning for rectangular regions. It can also be seen from the figures that the extension distance of scan-type path on both sides of the polygon is greater than the extension distance of the GPSA-GA. The main reason is that the VTOL UAV receives the influence of the turning radius when turning. When the interval between two adjacent paths in sequence is less than the turning radius, UAVs need to fly a long distance to turn to ensure that they can fly along the path.

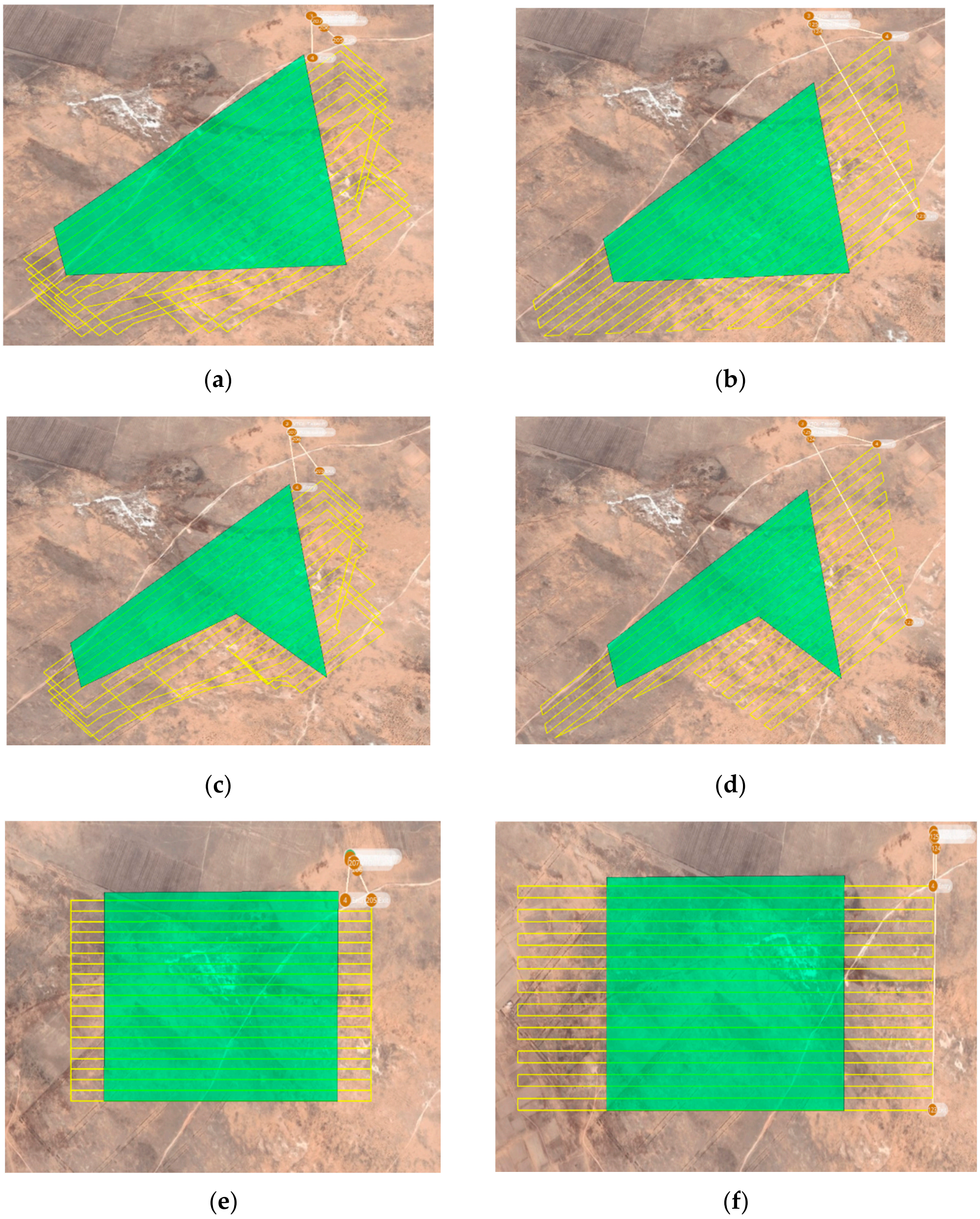
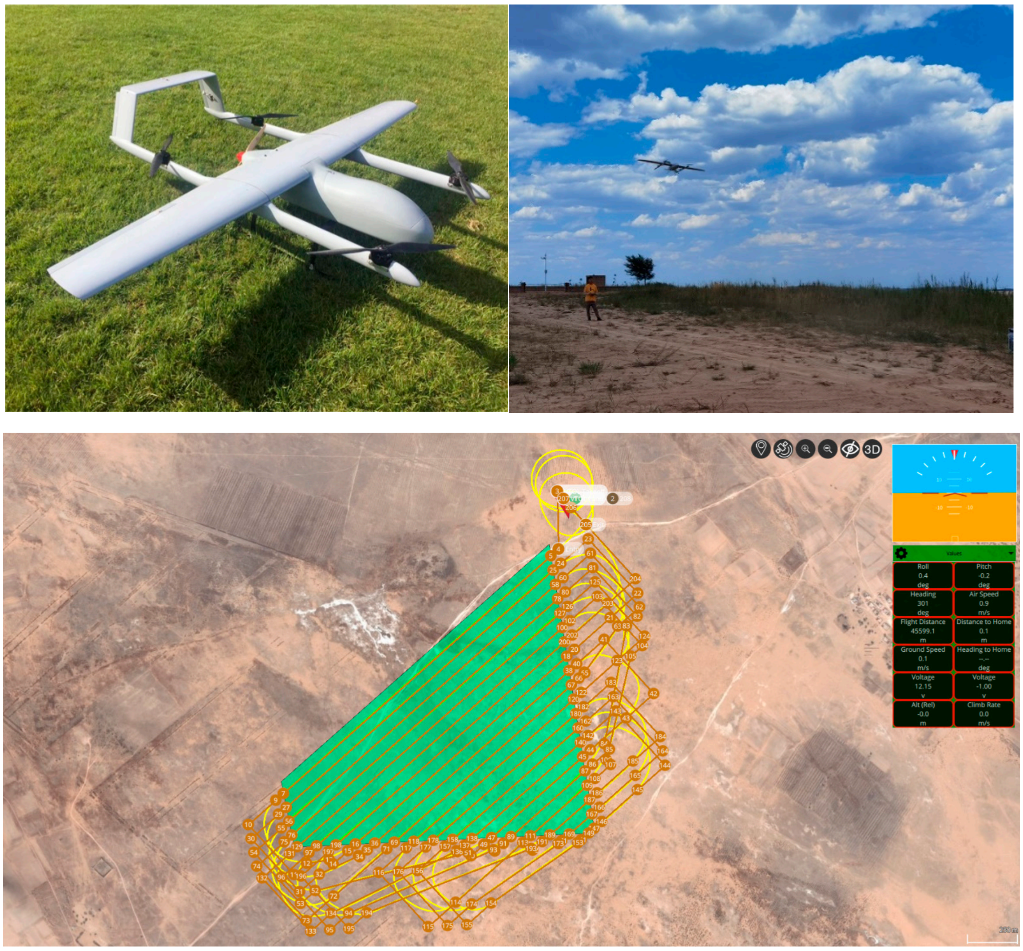


Figure 10. Path planning of different area shapes. (**a**) Convex polygon Area coverage using GPSA-GA, (**b**) Convex polygon Area coverage using Scan-type, (**c**) Concave polygon Area coverage using GPSA-GA, (**d**) Concave polygon Area coverage using Scan-type, (**e**) Rectangular area coverage using GPSA-GA, (**f**) Rectangular area coverage using Scan-type.

6.3. Flight Test

The actual flight test is conducted in a rectangular area. Three path planning methods—namely conventional GA, GPSA-AG, and Scan-type—are compared, where the first two methods are based on their optimal results from the previous algorithm tests. The flight height in the mission area is 300 m, the overlap rate 80%, the scan width is 300 m, and the regional path spacing 60 m. Figure 11 shows a photo of the studied VTOL UAV. Its maximum flight speed, minimum flight speed, and cruise speed are 25 m/s, 20 m/s, and 22 m/s, respectively. Total weight of the UAV is 14 kg. Roll angle (Φ) is 15 degrees. Its turning radius *R* is 180 m (In actual flight, the flight speed of the aircraft is about 21 m/s, so the minimum turning radius is lower than this value, but due to flight safety considerations, *R* = 180 m is selected according to the cruise speed). Due to the limitation of UAV flight time (2.5 h), we will only compare the proposed algorithm, GA, and the scanning algorithm under 20 and 50 paths, respectively. The results are shown in Figure 11. First, select the take-off and landing point (the same point), 80 m above which a flight mode switching point (to fixed-wing mode) is set. The distance from the take-off point to the entry point of the task area is 400 m. A circling point is set 300 m above ground, with a circling radius of 180 m. When the UAV hovers to the 300 m height, it flies to the task area to perform the area coverage. At the end of the task, it hovers and descends to 80 m high at the selected point, and then switches to the rotor mode to land.



Roll 0.4 deg
Pitch -0.2 deg
Heading 301 deg
Air Speed 0.9 m/s
Flight Distance 45599.1 m
Distance to Home 0.1 m
Ground Speed 0.1 m/s
Heading to Home --.-- deg
Voltage 12.15 v
Voltage -1.00 v
Alt (Rel) -0.0 m
Climb Rate 0.0 m/s

Figure 11. The VTOL UAV in flight test.

Fly-outs on the right in Figure 11 are enlarged views of the left screenshots, which also give the actual flight time and travel distance of the aircraft. The yellow line is the actual flight path, and the brown line is the planned path, also the extension distance required by the algorithm in this paper is shorter than that of the scanning algorithm, due to a wider spacing between the sequentially executed paths that is basically close to the turning radius of the UAV. The results of the actual flight are shown in Table 4.

Table 4. Flight test results.

Paths	Flight Time and Distance	Scan-Type	GA	GPSA-GA
20	Flight Time (min) Flight Distance (m)	52.43 59,779.6	38 43,096	37.25 42,520
50	Flight Time (min) Flight Distance (m)	129.53 146,394.2	86.07 99,585.8	85.03 98,882.9

With 20 paths, the total range covered by GPSA-GA is 42,520 m, whereas the counterpart of the Scan-type algorithm is 59,779.6 m, i.e., a saving of 17.26 km. With 50 paths, the two numbers are 98,882.9 m and 146,394.2 m, i.e., a saving of 47.51 km. Although only 20 and 50 paths are tested in the present study (limited by the endurance time of the aircraft, i.e., about 2.5 h), it is not hard to infer that even more range savings will be achieved for 80 and 100 paths. If path planning is carried out in the direction normal to the width of the polygon for the 50-path case, the planned area would be too large for the UAV to complete in one flight. Moreover, the flight area size is kept fixed for both algorithms for the same number of paths, in order to compare the two algorithms. In addition, GA has a shorter flight distance compared to GPSA-GA, but it is obviously better than scan-type algorithm. This is mainly because GA and GPSA consider the execution sequence of paths. The quality of the execution sequence has also been pointed out in algorithm testing several repeated calculations that GPSA-GA performs better than GA.

7. Conclusions

Coverage path planning is studied in this work. The regional CPP problem is first transformed into a TSP problem. The combination of good point set and genetic algorithm framework is employed to solve the problem. Simulations are carried out in scenarios of 20, 50, 80, and 100 covered paths. Compared to conventional GA, the proposed algorithm achieves better solutions and, at the same time, the improvement is more remarkable with the increase of the number of paths. This is also verified by the actual flight of 20 and 50 paths. However, increase in the number of paths also leads to the problem that the data size will be squared if the CPP problem is converted to a TSP problem, greatly reducing the convergence speed and increasing the corresponding planning time. In the future, region segmentation algorithms can be used to segment a large area, and algorithms with shorter planning time can be further studied.

In addition, this paper is only based on the coverage path planning of two-dimensional plane, which can be studied in three-dimensional space in the future to achieve better flight effect.

Author Contributions: Z.L. & Y.L.: Research funding, directing, Project administration and Resources. J.Y. & L.C.: Experiment execution, data processing, and manuscript writing. Q.A.: Research conceptualization and manuscript review. B.M. & L.W.: Algorithm formalization and testing. All authors have read and agreed to the published version of the manuscript.

Funding: This study is funded by National Natural Science Foundation Fund (No. 52072309), Key Research and Development Program of Shaanxi (Program No. 2019ZDLGY14-02-01), Shenzhen Fundamental Research Program (No. JCYJ20190806152203506), and Aeronautical Science Foundation of China (No. ASFC-2018ZC53026).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Wang, S.; Bai, Y.; Zhou, C. Coverage Path Planning Design of Mapping UAVs Based on Particle Swarm optimization Algorithm. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8236–8241.
2. Yao, P.; Xie, Z.; Ren, P. Optimal UAV Path Planning for Coverage Search of Stationary Target in River. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 822–829. [CrossRef]
3. Paull, L.; Thibault, C.; Nagaty, A.; Seto, M.; Li, H. Sensor-Driven Area Coverage for an Autonomous Fixed-Wing Unmanned Aerial Vehicle. *IEEE Trans. Cybern.* **2014**, *44*, 1605–1618. [CrossRef] [PubMed]
4. Guerrero, J.A.; Bestaoui, Y. UAV Path Planning for Structure Inspection in Windy Environments. *J. Intell. Robot. Syst.* **2013**, *69*, 297–311. [CrossRef]
5. Yongqiang, X.; Zhongbo, L.; Jin, Q.; Kai, Z. Optimal video communication strategy for intelligent video analysis in unmanned aerial vehicle applications. *Chin. J. Aeronaut.* **2020**, *33*, 101–109.

6. Sujit, P.B.; Hudzietz, B.P.; Saripalli, S. Path Planning for Angle Constrained Terrain Mapping Using an Unmanned Aerial Vehicle. *J. Intell. Robot. Syst.* **2013**, *69*, 273–283. [CrossRef]
7. Kerle, N.; Nex, F.; Gerke, M.; Duarte, D.; Vetrivel, A. UAV-Based Structural Damage Mapping: A Review. *Int. J. Geo-Inf.* **2019**, *9*, 14. [CrossRef]
8. Zhan, J.; Xie, W.; Guo, Q.; Zhang, P. An Improved UAV Coverage Search Path Planning Method. In Proceedings of the 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), Xiamen, China, 10–12 August 2018; pp. 1–5.
9. Niendorf, M.; Kabamba, P.T.; Girard, A.R. Stability of Solutions to Classes of Traveling Salesman Problems. *IEEE Trans. Cybern.* **2016**, *46*, 973–985. [CrossRef]
10. Babaie, O.; Esfahany, M.N. Optimization and Heat Integration of Hybrid R-HIDiC and Pervaporation by Combining GA and PSO Algorithm in TAME Synthesis. *Sep. Purif. Technol.* **2019**, *236*, 1–3. [CrossRef]
11. Tewolde, G.S.; Sheng, W. Robot Path Integration in Manufacturing Processes: Genetic Algorithm versus Ant Colony Optimization. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2008**, *38*, 278–287. [CrossRef]
12. Chuangling, L.; Huaiwang, L.; Jingyu, Y. A path planning method based on adaptive genetic algorithm for mobile robot. *J. Inf. Comput. Sci.* **2011**, *8*, 808–814.
13. Hong, Q.; Xing, K.; Takacs, A. An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing* **2013**, *120*, 509–517.
14. Lee, J.; Kim, D.W. An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph. *Inf. Sci.* **2016**, *332*, 1–18. [CrossRef]
15. Tuncer, A.; Yildirim, M. Dynamic path planning of mobile robots with improved genetic algorithm. *Comput. Electr. Eng.* **2012**, *38*, 1564–1572. [CrossRef]
16. Filho, A.L.; Vettorazzi, D.; Cruz, A.; Soares, A. ATM: A New Heuristic Algorithm Based on Genetic Algorithm and Betting Theory. *IEEE Lat. Am. Trans.* **2017**, *15*, 510–516. [CrossRef]
17. Holst, L.T. Genetic Algorithms Applied to Multi-Objective Aerospace Shape Optimization. *J. Aerosp. Comput. Inf. Commun.* **2005**, *2*, 217–235. [CrossRef]
18. Li, Y.; Wang, R.; Yu, L.; Xu, M. Satellite range scheduling with the priority constraint: An improved genetic algorithm using a station ID encoding method. *Chin. J. Aeronaut.* **2015**, *28*, 789–803. [CrossRef]
19. Liu, Y.; Jafari, S.; Nikolaidis, T. Advanced optimization of gas turbine aero-engine transient performance using linkage-learning genetic algorithm: Part II, optimization in flight mission and controller gains correlation development. *Chin. J. Aeronaut.* **2021**, *34*, 568–588. [CrossRef]
20. Wang, Z.; Cheng, J.; Yao, F. Solving of Flow Shop Scheduling Problem Based on Good Point Set Genetic Algorithm. *Comput. Eng.* **2004**, *30*, 10–13.
21. Pei, S.Y. Multi-objective Evolution Algorithm of Good Point Set for Solving Restraint Optimization Problem. *Comput. Eng.* **2011**, *37*, 153–154.
22. Wang, R.; Ru, Y.; Long, Q. Improved Adaptive and Multi-group Parallel Genetic Algorithm Based on Good-point Set. *J. Softw.* **2009**, *4*, 348–356. [CrossRef]
23. Cabreira, T.M.; Di Franco, C.; Ferreira, P.R.; Buttazzo, G.C. Energy-Aware Spiral Coverage Path Planning for UAV Photogrammetric Applications. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3662–3668. [CrossRef]
24. Lin, L.; Goodrich, M. Hierarchical Heuristic Search Using a Gaussian Mixture Model for UAV Coverage Planning. *IEEE Trans. Cybern.* **2014**, *44*, 2532–2544. [CrossRef]
25. Xie, J.; Garcia Carrillo, L.R.; Jin, L. Path Planning for UAV to Cover Multiple Separated Convex Polygonal Regions. *IEEE Access* **2020**, *8*, 51770–51785. [CrossRef]
26. Richards, A. Flight Optimization for an Agricultural Unmanned Air Vehicle. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; Volume 2018, pp. 2891–2896.
27. Liu, C.H.; Ma, X.; Gao, X.; Tang, J. Distributed Energy-Efficient Multi-UAV Navigation for Long-Term Communication Coverage by Deep Reinforcement Learning. *IEEE Trans. Mob. Comput.* **2020**, *19*, 1274–1285. [CrossRef]
28. Ait Bouziaren, S.; Aghezzaf, B. An Improved Augmented ε -Constraint and Branch-and-Cut Method to Solve the TSP with Profits. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 195–204. [CrossRef]
29. Janos, J.; Vonasek, V.; Penicka, R. Multi-Goal Path Planning Using Multiple Random Trees. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4201–4208. [CrossRef]
30. Heo, S.; Park, K.; Choi, E.-H.; Bien, F. Differential Coded Multiple Signaling Method with Fully Differential Receiver for Mutual Capacitive Fingerprint TSP. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2019**, *67*, 74–85. [CrossRef]
31. Qian, Y.; Yang, X.; Jing, W.; Zhang, W. An improved numerical method for constructing Halo/Lissajous orbits in a full solar system model. *Chin. J. Aeronaut.* **2018**, *31*, 1362–1374. [CrossRef]
32. Wang, L.; Huang, Z.; Yun, Z. Navigation message designing with high accuracy for NAV. *Chin. J. Aeronaut.* **2014**, *27*, 994–1001. [CrossRef]
33. Zuo, Z.; Mallikarjunan, S. L1 Adaptive Backstepping for Robust Trajectory Tracking of UAVs. *Ind. Electron. IEEE Trans.* **2017**, *64*, 2944–2954. [CrossRef]
34. Zhi-Jun, L. Immune good-point set genetic algorithm. *Comput. Eng. Appl.* **2007**, *43*, 37–40.

35. Zhang, L.; Zhang, B. Good Point Set Based Genetic Algorithm. *Chin. J. Comput.* **2001**, *24*, 917–922.
36. Zhao, C.; Ling, Z. A Good-point Set Genetic Algorithm for TSP. *Comput. Eng. Appl.* **2001**, *37*, 83–84.