

Article

Study on Multi-UAV Cooperative Path Planning for Complex Patrol Tasks in Large Cities

Hongyu Xiang , Yuhang Han, Nan Pan , Miaohan Zhang and Zhenwei Wang

Faculty of Civil Aviation and Aeronautical, Kunming University of Science and Technology, Kunming 650500, China

* Correspondence: nanpan@kust.edu.cn; Tel.: +86-15808867407

Abstract: Unmanned Aerial Vehicles (UAVs) are increasingly utilized for urban patrol and defense owing to their low cost, high mobility, and rapid deployment. This paper proposes a multi-UAV mission planning model that takes into account mission execution rates, flight energy consumption costs, and impact costs. A kinematics and dynamics model of a quadcopter UAV is established, and the UAV's flight state is analyzed. Due to the difficulties in addressing 3D UAV kinematic constraints and poor uniformity using traditional optimization algorithms, a lightning search algorithm (LSA) based on multi-layer nesting and random walk strategies (MNRW-LSA) is proposed. The convergence performance of the MNRW-LSA algorithm is demonstrated by comparing it with several other algorithms, such as the Golden Jackal Optimization (GJO), Hunter-Prey Optimization (HPO), Pelican Optimization Algorithm (POA), Reptile Search Algorithm (RSA), and the Golden Eagle Optimization (GEO) using optimization test functions, Friedman and Nemenyi tests. Additionally, a greedy strategy is added to the Rapidly-Exploring Random Tree (RRT) algorithm to initialize the trajectories for simulation experiments using a 3D city model. The results indicate that the proposed algorithm can enhance global convergence and robustness, shorten convergence time, improve UAV execution coverage, and reduce energy consumption. Compared with other algorithms, such as Particle Swarm Optimization (PSO), Simulated Annealing (SA), and LSA, the proposed method has greater advantages in addressing multi-UAV trajectory planning problems.

Keywords: lightning search algorithm; multi-layer nesting strategy; path planning; quadcopter UAV; urban patrol



Citation: Xiang, H.; Han, Y.; Pan, N.; Zhang, M.; Wang, Z. Study on Multi-UAV Cooperative Path Planning for Complex Patrol Tasks in Large Cities. *Drones* **2023**, *7*, 367. <https://doi.org/10.3390/drones7060367>

Academic Editor: Pablo Rodríguez-González

Received: 23 April 2023

Revised: 27 May 2023

Accepted: 28 May 2023

Published: 1 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the era of 5G and the upcoming 6G, the demand for UAVs in public safety and other fields is increasing. UAVs equipped with infrared thermal dual-light cameras play an important role in smart city aerial intelligence, providing a practical solution for large-scale urban patrols. Smart city aerial intelligence is a feasible and cost-effective solution to address the challenges of large inspection and supervision areas, heavy inspection tasks, and limited frontline inspection staff in the process of urban governance, reducing potential safety hazards and illegal incidents [1]. Multi-UAV coordinated trajectory planning for large urban patrols is crucial in the process of autonomous patrol tasks performed by multiple UAVs and directly impacts the efficiency of patrol missions. It is the key to autonomous patrol tasks performed by UAVs [2].

The use of UAVs encompasses a wide range of applications, such as monitoring tasks [3], logistics distribution [4], warehousing [5], and industrial [6] operations, among others. At this stage, although there has been much research on UAV patrols [7], efficient UAV patrol path-planning algorithms in cities are almost missing. Yao et al. [8] developed a new algorithm based on disturbing fluid and trajectory propagation to solve the problem of UAV's three-dimensional path planning in a static environment. Xia et al. [9] established

a trajectory optimization model based on uniform time intervals and proposed a gradient-based sequential minimum optimization (GB-SMO) algorithm to solve the UAV planning problem. Gao et al. [10] also studied trajectory planning for battlefield missions based on the war environment, based on the improved RRT algorithm, with the shortest trajectory and the shortest planning time as the goal. The above studies have carried out useful explorations on multi-UAV trajectory planning based on various backgrounds; however, none of them involve the field of urban patrol. Further, Wang et al. [11] proposed a heuristic algorithm for urban road network surveillance tasks to study multi-aircraft trajectory planning for rotary-wing UAVs. In addition, Zhang et al. [12], based on the urban mission background, aimed at the problem that UAV flying in the urban environment posed a greater threat to pedestrians and public property and established a mathematical model of UAV trajectory planning with the introduction of risk assessment. Further, they proposed an improved ant colony algorithm for optimal trajectory planning and achieved certain results. Na et al. [13], on the other hand, focused on 3D path optimization and real-time obstacle avoidance of UAVs based on the urban context. Muñoz et al. [14] instead focused on the formation control of UAVs in cities. Wang et al. [15] also studied UAV flight planning based on the urban environment and hybrid strategy to ensure the smooth progress of urban flight. In addition, there is still much research on UAV trajectory planning based on the urban environment [16,17]. The above studies have all made significant contributions to the field of urban patrol. However, their research objectives are limited to the shortest trajectory and the maximum safety factor, task completion rate, and other issues.

Involving the UAV trajectory planning algorithm, Dai et al. [18] used the genetic algorithm to plan the path of the UAV in small-scale trajectory planning problems. Yuan et al. [19] added pheromone and heuristic functions to the traditional particle swarm algorithm to realize the path planning of UAVs. He et al. [20] proposed a novel hybrid algorithm called HIPSO-MSOS by combining Improved Particle Swarm Optimization (IPSO) and Modified Symbiotic Search (MSOS). There are also methods based on the mixed integer linear programming (MILP) method [21], the hybrid genetic and A* algorithm [22], the improved CSA algorithm [23], the PIOFOA algorithm [24], etc. However, these methods are not applicable for multi-UAV trajectory planning under multiple constraints, as it is difficult to find an optimal solution. The biological heuristic algorithm is widely used in large-scale optimization problems, such as multi-aircraft cooperative trajectory planning, due to its excellent characteristics [25–27]. Sun et al. [28] used the bat algorithm to optimize UAV mission planning. Duan et al. [29] proposed a dynamic discrete pigeon-inspired optimization algorithm based on multi-UAV coordinated trajectory planning in a three-dimensional environment. Jain et al. [30] used the multi-universe algorithm to research UAV trajectory planning in a dynamic environment. The above research on various UAV trajectory planning algorithms has carried out a useful exploration to optimize UAV trajectory with certain reference significance. The lightning search algorithm [31], a new biological heuristic algorithm, has the advantages of a simple structure and a short calculation time. However, when applied to the three-dimensional trajectory planning of UAVs, the search accuracy is low, and it is easy to fall into optimal local values, resulting in the problem of algorithm premature convergence. Therefore, this paper aims to improve the shortcomings of the lightning search algorithm, which are that it is easy to fall into the local optimum and the uniformity is not good enough.

Based on the studies mentioned above, this paper aims to establish a multi-rotor UAV-coordinated mission planning model for urban patrol. The model incorporates a multi-objective function that considers the least impact cost of UAVs, the shortest UAV flight energy consumption cost, and the highest UAV mission execution rate. This paper introduces a multi-level nesting strategy and random walk strategy to improve the global search ability and convergence accuracy of the lightning search algorithm (LSA). Furthermore, the improved algorithm is combined with the RRT algorithm, which introduces the greedy strategy to overcome the problems of falling into the optimal local solution and slow convergence speed when solving the trajectory planning problem. Finally, a real patrol en-

environment based on a satellite image of a specific location in Beijing is established to verify the effectiveness of the proposed multi-rotor UAV urban patrol mission trajectory planning program. The simulation results show that the proposed algorithm has fast convergence speed and high accuracy compared with LSA and other algorithms. It can efficiently plan the trajectory for UAVs performing urban patrol tasks, thereby improving the urban patrol efficiency of the multi-rotor UAV. The UAV to perform patrol tasks schematic is shown in Figure 1.

The remainder of this paper is organized as follows. The urban environment model, UAV cost model, etc., are built in Section 2, modeling of UAV mission planning, which introduces the concept and necessity of urban patrol. Urban environment modeling establishes the environment model of the city. Trajectory cost modeling establishes the UAV cost model for urban patrol. Quadrotor UAV system modeling is built in Section 3, quadrotor UAV system modeling. The improvement strategy of the RRT algorithm and the process and improvement based on the lightning search algorithm is described in Section 4, algorithm description based on multi-layer nesting and randomized swimming strategy. The simulated data and graphs are shown in Section 5, simulation results. Simulation results conduct experimental simulations by the improved RRT algorithm. The conclusion draws conclusions. Finally, the conclusions and contributions of this paper based on the simulation results are in Section 6, conclusions and future research.

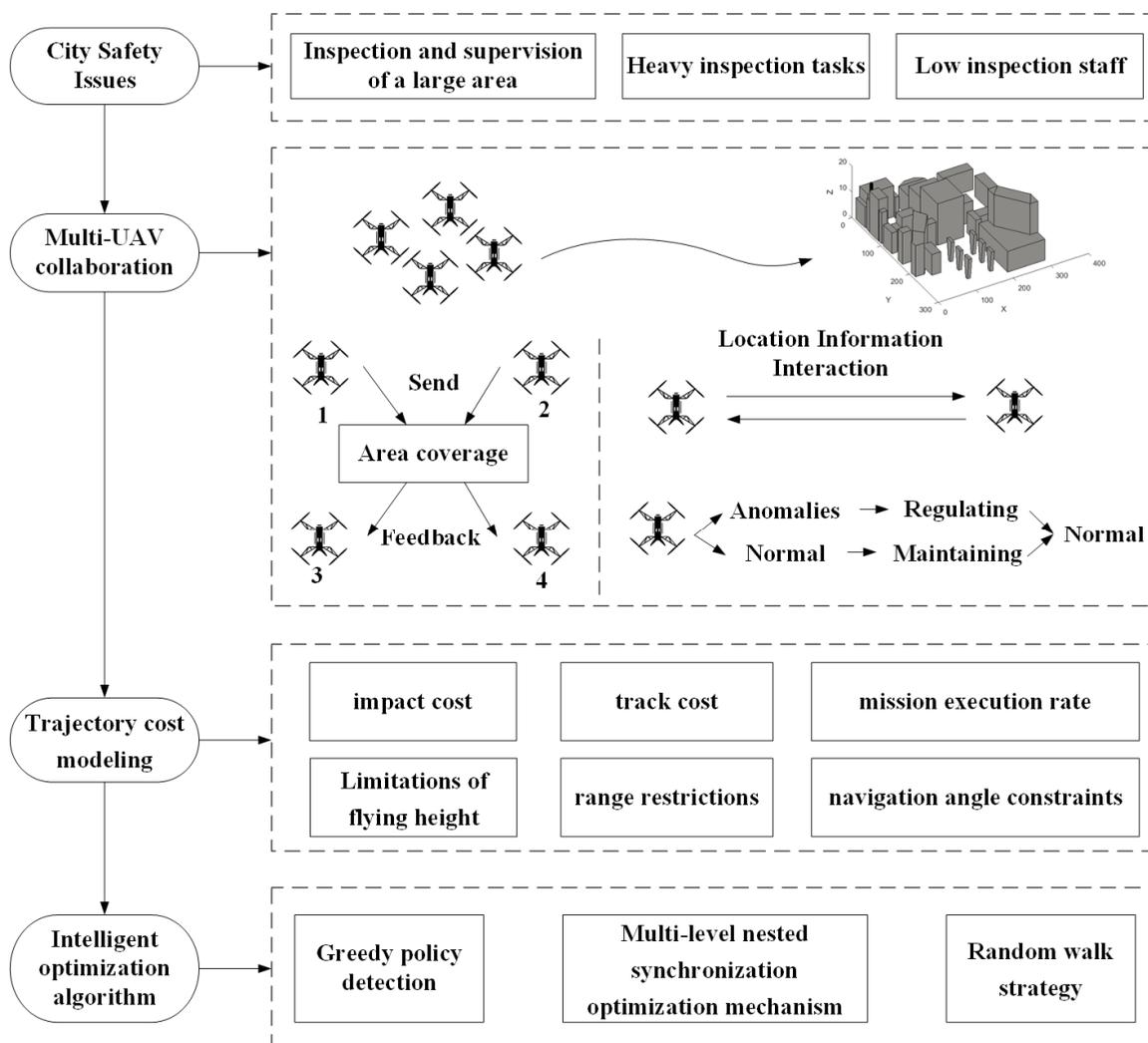


Figure 1. The UAV to perform patrol tasks schematic.

2. Modeling of UAV Mission Planning

In this section, the mission description of UAV city patrol is first introduced, then the environmental model is established, and a multi-UAV mission planning model that considers UAV mission execution rate, flight energy consumption cost, and impact cost are developed.

2.1. Problem Description

As smart cities move towards high-quality construction and development, the demand for intelligent information control is increasing. UAVs can conduct comprehensive surveys and maps of urban management effects at different times, providing the government with accurate and objective bases for urban management. The specific tasks of UAV patrol operations in large cities are as follows: Each UAV departs from the same center and returns to the center after completing all missions, flying according to the established planned route and not accepting new mission assignments. The urban environment is complex, with many pedestrians and vehicles, and a crash can cause damage to pedestrians and vehicles. Therefore, it is necessary to ensure that all patrol tasks are carried out under the premise of the minimum danger to the public in the event of a crash and with the shortest trajectory as the goal. Ultimately, a flight path with full coverage of the patrol area is planned for multiple UAVs.

2.2. Urban Environment Modeling

In a complex and dense urban environment, UAV patrols encounter a wide variety of obstacles, unevenly distributed and restricted by the complex constraints of the airspace environment. Three-dimensional environment modeling solves the prerequisite of UAV path planning. To improve the efficiency of UAV planning, create a simplified modeling diagram of the city's three-dimensional space. Simplify various obstacles into rectangular bodies, and perform environmental modeling after simplifying all buildings, just like Figure 2. All buildings are avoidance areas for UAVs to fly, and the plane coordinates of the γ -th building are expressed as $(x_1, y_1), (x_2, y_2), \dots, (x_{\beta_\gamma}, y_{\beta_\gamma})$. The height is expressed as h_γ , β_γ and it is the number of plane coordinates of γ building coordinates. Where $\gamma \in (1, n)$, n is the number of buildings. Avoidance area constraints of ground UAVs are

$$\sum_{\gamma=1}^n \left(\begin{array}{c} (x_1, y_1, 0), (x_2, y_2, 0), \dots, (x_{\beta_\gamma}, y_{\beta_\gamma}, 0) \\ (x_1, y_1, h_\gamma), (x_2, y_2, h_\gamma), \dots, (x_{\beta_\gamma}, y_{\beta_\gamma}, h_\gamma) \end{array} \right), \quad (1)$$

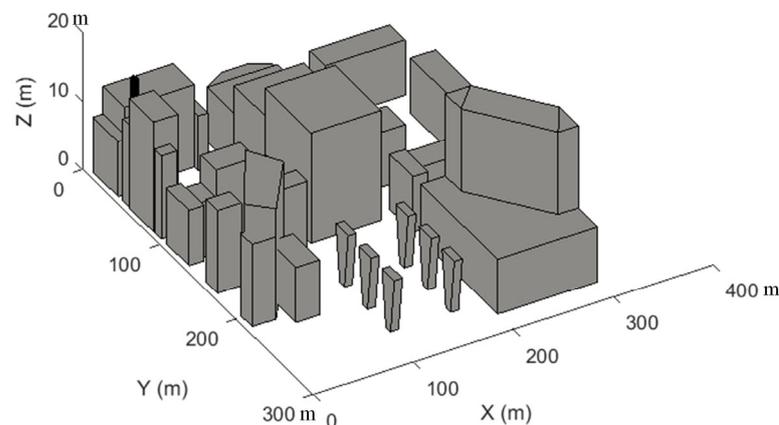


Figure 2. Example map of urban 3D environment modeling.

2.3. Trajectory Cost Modeling

Building upon the mathematical model proposed by Zhang et al. [8], this paper introduces a cost-effectiveness ratio of UAVs equipped with infrared thermal dual-light cameras to enhance urban patrols' efficiency and minimize energy consumption. It should

be noted that the assumptions in this study are limited to urban architectural environments and do not account for geographical features, such as mountains and basins. The UAV impact cost formula is presented below:

$$R = f_r CT + f_r \rho_P \frac{h}{H_{\max}}, \quad (2)$$

The UAV impact cost formula includes several variables. R represents the impact cost of the UAV in case of a fall. f_r denotes the reliability factor of the UAV. C reflects the density of vehicles per unit area while ρ_P represents the density of population per unit area. T indicates the death rate of the population due to car accidents. H_{\max} and h are the highest-flying height and true flying height of the UAV, respectively.

The formula for UAV flight energy consumption cost with e representing the energy consumed per unit distance of flight is presented below:

$$E_C = e \sum \left(\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \right), \quad (3)$$

2.4. UAV Mission Execution Rate

The infrared thermal dual-lens camera is equipped with a 25 mm lens, 90° infrared light, and a shooting angle of 90°, with the best observation distance being between 50–100 m. The objective is to maximize the patrol area of each UAV while minimizing the repetition rate of patrols. The patrol area of each UAV is presented below:

$$S_i = 2l_i h_i \tan\left(\frac{\alpha}{2}\right), \quad (4)$$

where l_i is the track length of each UAV. h_i is the flying height of each UAV. S is the maximum coverage angle that the UAV equipped with the camera can shoot on the vertical plane. Where l_i represents the track length of each UAV, h_i denotes the flying height of each UAV, and α indicates the maximum coverage angle that the camera-equipped UAV can capture on the vertical plane.

$$S = \sum S_i, \quad (5)$$

$$P_{i,i+1} = \begin{cases} 0, & u \geq 2u_0 \\ \frac{2u_0 - u}{2u_0}, & u < 2u_0 \end{cases} \quad (6)$$

where $P_{i,i+1}$ represents the patrol repetition rate of the two UAVs, u is the flight distance between the two UAVs, and u_0 is defined as the minimum distance between any two UAVs to avoid repeated patrols.

$$P = \sum P_{i,i+1}, \quad (7)$$

Therefore, the objective function is as follows:

$$\min f = \frac{RE_C P}{S}, \quad (8)$$

This is expressed as the lowest impact cost and UAV flight energy consumption cost, the largest patrol area, and the lowest UAV patrol repetition rate in the route from departure to return when multiple UAVs are dispatched for urban patrol missions.

2.5. Restrictions

Based on the three-dimensional city map established in Section 2 and the trajectory objective function established in the previous section, the following constraints are formulated.

UAV flight range constraints. UAV patrols are restricted by relevant legal supervision and other factors, and there are space restrictions on patrol areas. Let x_{\min} , x_{\max} , y_{\min} , y_{\max} ,

z_{\min} , and z_{\max} be the minimum and maximum spatial locations that the relevant laws allow UAVs to reach. The constraint is expressed as the following:

$$\begin{cases} x_{\min} \leq x_i \leq x_{\max} \\ y_{\min} \leq y_i \leq y_{\max} \\ z_{\min} \leq z_i \leq z_{\max} \end{cases}, \tag{9}$$

where x_{\min} , x_{\max} , y_{\min} , y_{\max} , z_{\min} , and z_{\max} are the space limits for UAVs to perform patrol missions.

Limitations of UAV flying height. There exists a minimum flight height h_{\min} and a maximum flight height h_{\max} for UAV manufacturing specifications to ensure flight safety when performing patrol tasks, which are constrained to be expressed as the following:

$$h_{\min} \leq h_i \leq h_{\max}, \tag{10}$$

UAV range restrictions. The energy carried by UAVs is limited, and there is a maximum range limit. Let E_{\max} be the maximum energy the UAV possesses before taking off. The constraint is expressed as the following:

$$E_c \leq E_{\max}, \tag{11}$$

Patrol area coverage limit. When the UAV performs a patrol mission, subject to the mission requirements, the UAV patrol coverage area S needs to be larger than the other mission requirements.

According to the patrol mission requirements, the minimum patrol area is defined as S_c . The corresponding constraint can be expressed as the following:

$$S \geq S_c, \tag{12}$$

UAV navigation angle constraints are shown below:

$$\beta_i = \cos^{-1} \left[\frac{\overline{x_{i+1}x_i x_{i-1}} + \overline{y_{i+1}y_i y_{i-1}}}{\sqrt{\overline{x_{i+1}x_i^2 + x_i x_{i-1}^2}} \sqrt{\overline{y_{i+1}y_i^2 + y_i y_{i-1}^2}}} \right] \tag{13}$$

$(z_i = z_{i+1})$

$$0 \leq \beta_i \leq \beta_{\max}, \tag{14}$$

$$\mu_i = \tan^{-1} \left[\frac{|z_i - z_{i-1}|}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \right] \tag{15}$$

$(z_i \neq z_{i+1})$

$$0 \leq \mu_i \leq \mu_{\max}, \tag{16}$$

In Equation (13), β_i is the turning angle at position (x_i, y_i, z_i) . When the UAV performs a turning operation, it satisfies $z_i = z_{i+1}$. Equation (14) is the turning angle constraint of the UAV, where β_{\max} is the maximum turning angle of the UAV. In Equation (15), μ_i is the climb angle at position (x_i, y_i, z_i) . When the UAV is performing a climb operation, $z_i \neq z_{i+1}$. Equation (16) is the climb angle constraint of the UAV, where μ_{\max} is the maximum climb angle of the UAV.

3. Quadrotor UAV System Modeling

In this section, kinematic models and kinematic models are established for the flight state of quadrotor UAVs to conduct mechanical analysis of UAVs in performing tasks and ensure the integrity and science of UAV mission planning.

3.1. Quadrotor UAV Kinematic Model

$O_g X_g Y_g Z_g$ denotes the inertial coordinate system of the UAV, $O_b X_b Y_b Z_b$ denotes the body coordinate system of the UAV, and $P_g = [x, y, z]^T$ is the position vector of the UAV in the inertial coordinate system with roll angle φ , pitch angle θ , and deflection angle ψ . The rotation matrix from the body coordinate system to the inertial coordinate system is R_b :

$$R_b = \begin{bmatrix} \cos \theta \cos \psi - \sin \varphi \sin \theta \sin \psi & -\cos \varphi \sin \psi & \sin \theta \cos \psi + \sin \varphi \cos \theta \sin \psi \\ \cos \theta \sin \psi + \sin \varphi \sin \theta \cos \psi & \cos \varphi \cos \psi & \sin \theta \sin \psi - \sin \varphi \cos \theta \cos \psi \\ -\cos \varphi \sin \theta & \sin \varphi & \cos \varphi \sin \theta \end{bmatrix}, \quad (17)$$

The velocity vector of the UAV center of mass at $O_g X_g Y_g Z_g$ is denoted as v_g and the angular velocity vector is Ω , and the roll angle φ , pitch angle θ , and deflection angle ψ are denoted by Θ .

$$\Theta = [\varphi, \theta, \psi]^T, \quad (18)$$

The relationship between the angular velocity vector in the inertial coordinate system and the three angles of roll angle $\dot{\Theta} = W\Omega$, pitch angle φ , and deflection angle θ is expressed by the equation ψ .

$$W = \begin{bmatrix} \cos \theta & 0 & -\cos \varphi \sin \theta \\ 0 & 1 & \sin \varphi \\ \sin \theta & 0 & \cos \varphi \cos \theta \end{bmatrix}, \quad (19)$$

Define the augmented state variables of the UAV system as

$$q = [P_g, \Theta, \alpha]^T \in R^{6+p}, \quad (20)$$

where $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_p]^T \in R^p$, α_i are the angles i of the joints of the UAV system, and p are the degrees of freedom of the UAV system so that the kinematic equations of the UAV are

$$\dot{P}_g = [I_{3 \times 3}, O_{3 \times 3}, O_{p \times p}] \dot{q}, \quad (21)$$

$$\Omega = [O_{3 \times 3}, W, O_{p \times p}] \dot{q}, \quad (22)$$

where, $I_{n \times n}$ denotes the $n \times n$ -dimensional unit matrix and $O_{n \times n}$ denotes the $n \times n$ -dimensional zero matrix. The quadrotor UAV system and coordinate system are shown in Figure 3.

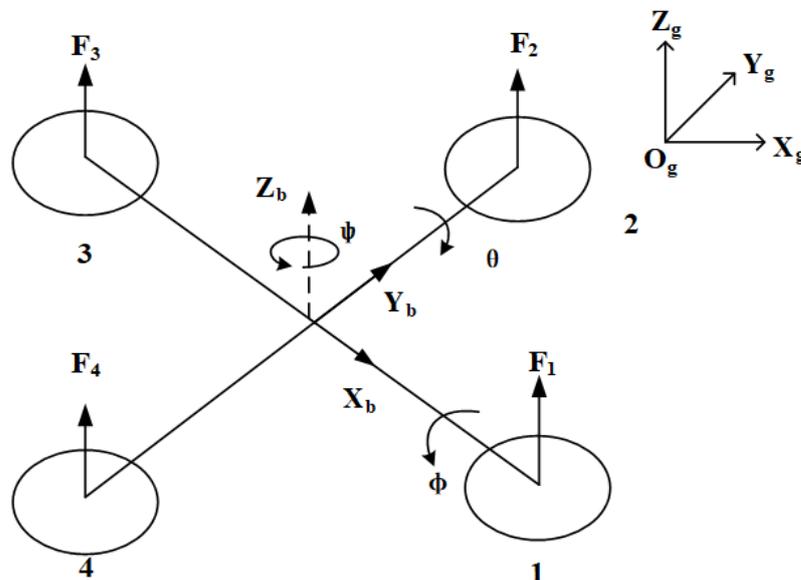


Figure 3. Quadrotor UAV system and coordinate system.

3.2. Quadrotor UAV Dynamics Model Building

Transformation matrix $R(\Theta)$, $R(\Theta) = R_z(\psi)R_y(\theta)R_x(\varphi)$ for UAV conversion from airframe coordinate system to geodetic coordinate system.

$$R_z(\psi)R_y(\theta)R_x(\varphi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{bmatrix}, \quad (23)$$

$$R(\Theta) = \begin{bmatrix} \cos \theta \cos \psi & \cos \psi \sin \theta \sin \varphi - \sin \psi \cos \varphi & \cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \\ \cos \theta \sin \psi & \sin \psi \sin \theta \sin \varphi + \cos \psi \cos \varphi & \sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix}, \quad (24)$$

Based on Newton's second law and the law of angular motion, it follows that

$$F = \frac{d}{dt}(mv) = ma, \quad (25)$$

$$\tau = I\dot{\omega}, \quad (26)$$

where m is the mass of the UAV, a is the linear acceleration of the UAV, F is the combined force of gravity and tension of the UAV, τ is the torque of the UAV, I is the rotational inertia of the UAV, and ω is the attitude angular velocity of the UAV.

Substituting the relationship between the lift and gravity of the UAV into Equation (25), the x , y , z directions of the UAV are represented in the inertial coordinate system as

$$R \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}, \quad (27)$$

$$\ddot{x} = \frac{f}{m}(\cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi), \quad (28)$$

$$\ddot{y} = \frac{f}{m}(\sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi), \quad (29)$$

$$\ddot{z} = \frac{f}{m}(\cos \theta \cos \varphi) - g, \quad (30)$$

The Eulerian angular moment of a UAV can be obtained from the Eulerian metric equation $M_T = I\dot{\omega} + \omega \times I\omega$ for a rigid body as

$$\begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \quad (31)$$

$$\ddot{\varphi} = \tau_\varphi / I_{xx} + \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) / I_{xx} = \frac{\sqrt{2}}{2}dc_T(w_1^2 - w_2^2 - w_3^2 + w_4^2) / I_{xx}, \quad (32)$$

$$\ddot{\theta} = \tau_\theta / I_{yy} + \dot{\varphi}\dot{\psi}(I_{zz} - I_{xx}) / I_{yy} = \frac{\sqrt{2}}{2}dc_T(w_1^2 + w_2^2 - w_3^2 - w_4^2) / I_{yy}, \quad (33)$$

$$\ddot{\psi} = \tau_\psi / I_{zz} + \dot{\varphi}\dot{\theta}(I_{xx} - I_{yy}) / I_{zz} = c_M(w_1^2 - w_2^2 + w_3^2 - w_4^2) / I_{zz}, \quad (34)$$

Therefore, the lift force of the UAV can be expressed as

$$f = c_T(w_1^2 + w_2^2 + w_3^2 + w_4^2), \quad (35)$$

$$\begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} c_T & \sqrt{2}dc_T & \sqrt{2}dc_T & c_M \\ c_T & -\sqrt{2}dc_T & \sqrt{2}dc_T & -c_M \\ c_T & -\sqrt{2}dc_T & -\sqrt{2}dc_T & c_M \\ c_T & \sqrt{2}dc_T & -\sqrt{2}dc_T & c_M \end{bmatrix} \begin{bmatrix} f \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}, \tag{36}$$

where, $w_i (i = 1, 2, 3, 4)$ denotes the rotational speed of the four rotors of the UAV, c_T denotes the constant, d is the distance between the center of the UAV and the center of the motor, I_{xx}, I_{yy}, I_{zz} denotes the rotational inertia in the direction of the x, y, z position, $\tau_\varphi, \tau_\theta, \tau_\psi$ denotes the ψ roll angle moment, pitch angle moment, yaw angle moment, and c_M is the constant.

4. Algorithm Description

This section proposes two improvements to the RRT algorithm for UAV trajectory planning. Firstly, the greedy strategy is integrated into the RRT algorithm. Secondly, the lightning search algorithm is enhanced by introducing multi-layer nesting and random walk strategy, which is combined with the improved RRT algorithm for UAV mission planning.

4.1. RRT Algorithm Optimized by Greedy Strategy (GP-RRT)

The RRT algorithm is currently considered the most advanced method for UAV path planning. It is particularly effective in navigating unexplored areas with ample space and few obstacles. However, due to its exhaustive search approach, it may not be suitable for directing UAVs to specific points. Therefore, to improve exploration efficiency and avoid blind search, we propose a greedy strategy that modifies the exploration path. This approach reduces exploration time, shortens the search path, and eliminates the search for unproductive blank areas.

4.1.1. Impact Checking

To prevent UAVs from colliding with 3D terrain, set the collision detection rules according to the terrain as follows:

$$\begin{cases} true, x_{new} \sim x_{arc} \\ false, else \end{cases}, \tag{37}$$

where the output is *true* for no collision and *false* for a collision with the building. x_{new} represents a new location node and x_{arc} represents a set of location points where the building is located.

4.1.2. Greedy Policy Detection

Simulation experiments are required to solve and validate the established model. The model includes limits on the mission execution rate and calculates the patrol repetition rate between UAVs. An optimization algorithm is used to obtain the optimal path of the UAV. In the context of RRT, if the initial path is generated using the traditional algorithm, the direction of optimization can be too large, making it difficult to quickly and efficiently find a better path. To address this issue, we propose using the sampling points of the target paranoid function for optimization while also adjusting the probability of the new node location to avoid invalid searches. In addition to the greedy strategy, we also incorporate a local expansion mechanism. This mechanism involves removing new nodes within the already traversed area in the existing node region. By doing so, we further enhance the optimization of the greedy strategy. The formula for updating nodes is as follows:

$$\begin{cases} x_{tree} = x_{new}, difference(x_{father}, x_{goal}) > difference(x_{rand}, x_{goal}) \\ x_{tree} = x_{father}, else \end{cases}, \tag{38}$$

The improvement strategy steps are as follows.

Step 1: randomly expand the tree to generate a new node x_{new} .
 Step 2: perform collision detection.
 Step 3: If there are no obstacles, perform greedy policy detection. If the greedy rule is met, add x_{new} to x_{tree} . If the greedy rule is not met, return to Step 1.
 Step 4: If there is an obstacle, connect x_{new} to the parent node x_{father} . Divide the line segment into 2 equal parts, mark it as x_{pre} , and return to Step 2.
 Step 5: Repeat Steps 1–4 until reaching the target point.
 The output includes the detection results for collision detection $state$, the result of collision detection $state_{CD}$, and the result of greedy detection $state_G$, as well as the number of iterations n . The pseudocode of the algorithm is shown in Figure 4.

```

Input :  $x_{goal}, x_{init}$ 
Result : A Path T from  $x_{init}$  to  $x_{goal}$ 
for i = 1 to n do
   $x_{rand} \leftarrow$  Choose Target( $x_{init}, x_{goal}$ );
   $state_{CD} \leftarrow$  Collision Detection( $x_{rand}, x_{father}$ );
while state == true
   $x_{pre} \leftarrow$  Equally Divide( $x_{new}, x_{father}$ );
   $state_{CD} \leftarrow$  Collision Detection( $x_{rand}, x_{father}$ );
else
   $state_G \leftarrow$  Greedy( $x_{new}, x_{goal}, x_{father}$ );
  while  $state_G ==$  true
     $x_{tree} \leftarrow$   $x_{new}$ ;
  else
    break
  end
end
if  $x_{new} == x_{goal}$  then
  success();

```

Figure 4. Pseudocode of GP-RRT.

The schematic diagram of the process is shown in Figure 5.

4.2. Lightning Search Algorithm Based on Multi-Level Nesting and Random Walk Strategy (MNRW-LSA)

In the field of multi-UAV path planning, there exist numerous optimization algorithms to solve such problems. However, the key challenges in the UAV trajectory cost model are the coverage rate and repetition rate, which are more difficult to address. Multi-UAV collaborative patrol tasks are complex and involve issues such as information interaction and path prioritization. It is essential for the individuals involved to possess diverse properties to enhance the effectiveness of the optimization algorithm. We suggest utilizing an enhanced version of the lightning search algorithm (LSA) to resolve the model. LSA mimics the natural occurrence of lightning discharge in the environment. According to the lightning conduction mechanism, there are three types of discharges: transient, spatial, and guided. Each type has unique discharge characteristics, and their probabilistic and detour characteristics can be utilized to optimize individual space.

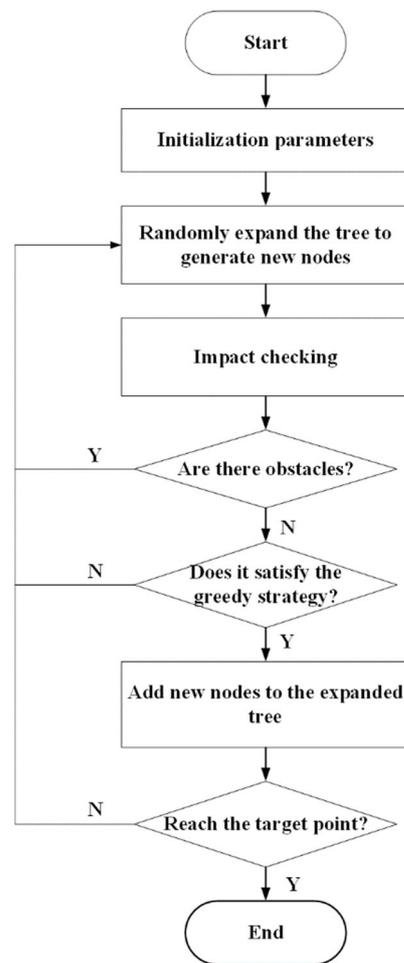


Figure 5. UAV path coding–decoding diagram.

This paper introduces a solution to the problem of weak random performance and low traversal coverage in the spatial search of an individual gradient optimization mechanism. The proposed solution involves implementing random touring based on the original algorithm and performing multi-layer nesting operations to achieve end re-optimization. A lightning search algorithm based on multi-layer nesting and random touring strategy is proposed as a way to complete the solution of the model.

4.2.1. Algorithm Principle

Transient Discharger

A transient discharger is a random-direction discharge body. Therefore, it can be considered a random number obtained from the standard uniform probability distribution on the open interval of the solution space—these dischargers whose positions satisfy the problem to be optimized. Generate an initial population that obeys a uniform distribution, and the population size is N . The probability density function of standard uniform distribution:

$$f(x^T) = \begin{cases} 1/b - a, & a \leq x^T \leq b \\ 0, & \text{otherwise} \end{cases} \quad (39)$$

where x^T is a random number that can provide a candidate solution. a and b are the lower and upper bounds of the solution space.

Spatial Discharger

Transition dischargers form the next channel. Let the position of the spatial discharger be $P^S = \{P_1^S, P_2^S, \dots, P_N^S\}$. Randomly generated numbers using an exponential distribution function with an exponential distribution probability density function:

$$f(x^S) = \begin{cases} \lambda e^{-\lambda x^S}, & x^S \geq 0 \\ 0, & otherwise \end{cases} \tag{40}$$

x^S represents the position of the space discharge body and represents the solution in the calculation formula. The position of the spatial discharger or the direction of the next iteration can be controlled by the shape parameter λ . P_i^S at the $i + 1$ iteration position can be described as the following:

$$P_{i_new}^S = P_i^S \pm e^{rand(\lambda_i)}, \tag{41}$$

where $e^{rand(\lambda_i)}$ is exponential random numbers, and if P_i^S is negative, then the resulting random number should be subtracted because Equation (40) only provides positive values.

However, the new position $P_{i_new}^S$ does not guarantee channel formation unless $P_{i_new}^S$ is a better solution. If $P_{i_new}^S$ provides a better solution in the next step, then P_i^S will be updated to $P_{i_new}^S$, otherwise it will remain unchanged. If $P_{i_new}^S$ is better than the current iteration, then the spatial discharger becomes guide discharger.

Guide Discharger

Randomly generated numbers using a normal distribution with a normal distributions probability density function:

$$f(x^L) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x^L-\mu)^2}{\sigma^2}}, \tag{42}$$

The randomly generated guide discharger can be searched in all directions from the current position defined by the shape parameter, and its mining capacity can be defined by the scale parameter. μ and σ represent the shape parameter and scale parameter in the normal distribution, respectively. The shape parameter σ of the induced discharge P^L exponentially decreases as it advances toward the Earth or finds the optimal solution. The position of the guide discharge P^L at the $i + 1$ iteration can be described as the following:

$$P_{i_new}^L = P_i^L + normrand(\mu^L, \sigma^L), \tag{43}$$

The basic lightning search algorithm steps are as follows.

Step 1: initialize the parameters, and set the number of races N , the maximum number of iterations M , and channel time T .

Step 2: initialize the transition discharger and calculate the fitness value of the discharger.

Step 3: update the fitness value of the guide discharger.

Step 4: update the best and worst transition dischargers and record the channel time.

Step 5: if the maximum channel time is reached, the worst channel is eliminated, and the channel time is reset to zero.

Step 6: update the conduction time and the fitness value of the discharger, and generate the spatial discharger and the guide discharger.

Step 7: update the transition discharger.

Step 8: If the maximum number of iterations is reached, the best discharge body is output; if the maximum number of iterations is not reached, return to Step 2.

4.2.2. Multi-Level Nested Synchronization Optimization Mechanism

In this study, we propose a multi-layer nested synchronization optimization mechanism to address the challenge of optimizing the paths of multiple UAVs during collaborative patrols. While it is relatively easy for UAVs to exchange information with each other, optimizing the coverage, cost, and repetition rate of each UAV patrol is a complex task. Our approach considers these factors and aims to find an optimal solution. The algorithm has been optimized by storing the optimized and unoptimized parts in separate spaces. This approach is superior to traditional optimization methods and makes the algorithm more effective in solving the model.

In the above at Step 5, if the renewal of the transient discharge body is not completed within the maximum conduction time, the time channel is eliminated. However, at the maximum conduction time, the transient discharge body of this time channel is superior to that of any other time channel. This paper utilizes a multi-level nesting operation in Step 5 to enhance the performance of the algorithm to prevent the loss of a more efficient discharge channel.

The implementation steps are as follows.

Step 1: put the location, X , of the discharge body in the time channel eliminated in Step 5 into the additional generated storage space.

Step 2: Randomly select the position of the discharge body in other time channels. If its position is better than x , the time channel will be eliminated.

Step 3: If x is better than the randomly selected discharge body position, the time is reset to zero, but the time channel is not eliminated.

The algorithm's pseudo code is shown in Figure 6.

```

Input : N, M, T
Result : fit, curve
for i = 1 to n do
  Dpointnew ← fitness(xinit)
  xb ← Preferential(Dpointnew)
  if t >= T
    xbad ← Area(xrand);
    if xrand >= xbad
      xbad ← NaN;
    else
      [Area] ← xbad;
    end
    t ← 0;
  else
    fitEd = fitness(xb);
    fitEp = fitness(xEp);
    if Ep < Ed
      xcurrent = xb;
    else
      stateB = Bifurcation();
      if stateB == ture
        [P1,P2] ← Passageway(stateB);
        Bad(P1,P2) ← NaN;
        xnew ← Best(P1,P2);
      end
    end
  end
  if m == M then
    success();
  end

```

Figure 6. Pseudocode of MNRW-LSA.

4.2.3. Random Walk Strategy

After obtaining the initial path using the improved RRT algorithm, direct path optimization often leads to duplicated patrol paths and a high repetition rate. However, utilizing the random touring strategy can effectively expand the search space and increase path diversity while still optimizing the path.

The discharge body obtained in Steps 5–7 does not have a definite time channel. In Step 8, if the maximum number of iterations is not reached, the conduction time will be increased, and the next cycle will be entered. This will disperse the positions of the better individuals. This paper proposes using a random walk strategy to concentrate the better discharge bodies and optimize their channels to address this. The set with the largest number of optimal discharge bodies is the most optimized area.

The position of the random walk strategy is updated as follows:

$$x'_i(m) = x_i(m) + \varepsilon(x_j(m) - x_k(m)), \tag{44}$$

where $x_i(m)$ is the i -th solution of the m -th generation. $x_j(m)$ and $x_k(m)$ are two random solutions of the m -th generation. m is the scaling factor, and its range is $\varepsilon \sim U(0, 1)$.

The flow chart of the MNRW-LSA algorithm is shown in Figure 7.

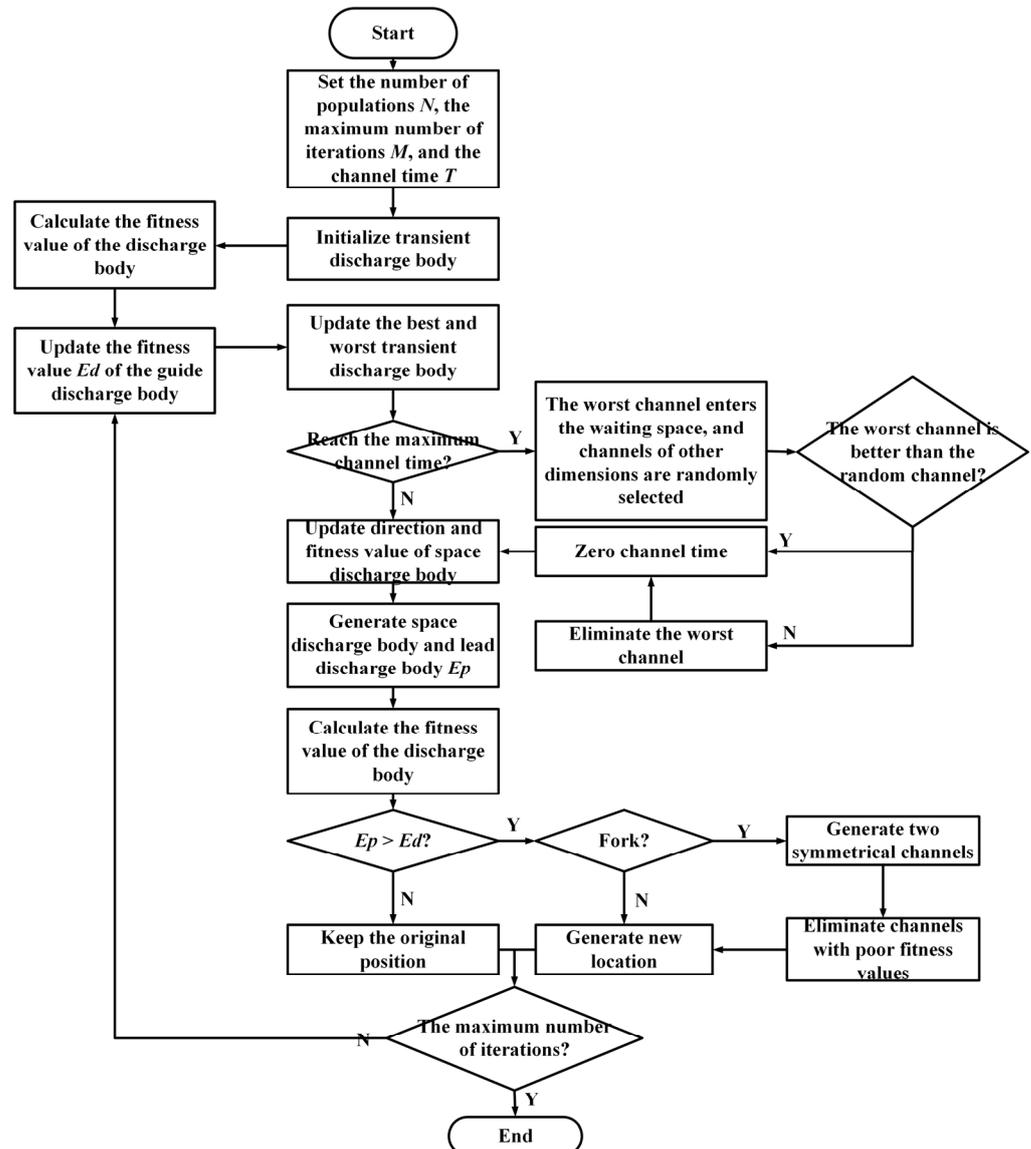


Figure 7. MNRW-LSA algorithm flow chart.

5. Simulation Results

In this section, the proposed MNRW-LSA algorithm is tested with eight benchmark functions and Friedman and Nemenyi tests. Moreover, a validity test of the GP-RRT algorithm was also conducted. Finally, the proposed model is simulated by the improved algorithm and compared with multiple algorithms, then further analyzed and verified according to the experimental results.

5.1. Algorithm Improvement Strategy Verification and Multi-Algorithm Data Comparison Analysis

First, to evaluate the improvement of the LSA by the strategy proposed in this paper, we first selected several typical test functions for comparison experiments, comparing GJO, HPO, POA, RSA, and LSA with the MNRW-LSA proposed in this paper, where the dimensionality is $\text{Dim} = 30$. The objective functions for conducting the tests and testing standards are shown in Tables 1 and 2 [32–35].

Table 1. Benchmark function.

Function
$F_1(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$
$F_2(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_1x_2)}{b_i^2 + b_1x_3 + x_4} \right]^2$
$F_3(x) = 4x_1^2 - 2 \cdot 1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
$F_4(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$
$F_5(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \right]$
$F_6(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^3 a_{ij} (x_j - p_j)^2 \right)$
$F_7(x) = - \sum_{i=1}^5 \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$
$F_8(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$

Table 2. Composite benchmark test functions.

Function	Dim	Range	Minimum
F_1	2	[−65, 65]	1
F_2	4	[−5, 5]	0.1484
F_3	2	[−5, 5]	−1
F_4	2	[−5, 5]	0.3
F_5	2	[−2, 2]	3
F_6	3	[1, 3]	−4
F_7	4	[0, 10]	−10
F_8	30	[−32, 32]	0

Figure 8 represents the search space range of the eight composite benchmark test functions, the fitness evolution curve of the MNRW-LSA algorithm, and the other four latest algorithms published in 2022. The 3D search space range visually reflects the solution value region within the range of values, and the curve plots clearly compare the performance of the algorithms. It can be seen from Table 3 that the MNRW-LSA algorithm proposed in this paper achieves the optimal solution value six out of eight times with the test results of different composite benchmark test functions. The results show the optimization performance of the MNRW-LSA algorithm in the composite function and prove that the MNRW-LSA algorithm can be used in other optimization problems.

Table 3. Comparison table of test results.

Functions	Theoretical Minimum Value	Optimization Result Value						Is the MNRW-LSA Algorithm Optimal?
		GJO	HPO	POA	RSA	LSA	MNRW-LSA	
F_1	1	0.998	0.998	0.998	11.14	0.998	12.67	0
F_2	0.1484	0.0003483	0.0003133	0.0003075	0.09601	0.001293	0.002039	1
F_3	-1	-1.032	-1.032	-1.032	-1.032	-1.032	-1.017	1
F_4	0.3	0.3981	0.3979	0.3979	0.3979	0.4	0.3979	1
F_5	3	3	3.003	3	12.38	3	3	1
F_6	-4	-3.855	-3.855	-3.863	-3.863	-3.754	-3.581	1
F_7	-10	-10.11	-10.15	-5.055	-10.15	-5.055	-9.953	1
F_8	0	8.882×10^{16}	4.441×10^{15}	8.882×10^{16}	4.441×10^{15}	8.882×10^{16}	20.22	0

5.2. Friedman and Nemenyi Test

In order to present a comprehensive statistical analysis of the comparison of optimization algorithms based on a non-parametric test of significance, we conducted a study for the eight typical benchmark test functions proposed above. The analysis of the results is completed with the variation of each test function in three different dimensions (50, 100, and 200) and clearly illustrates how the algorithm proposed in this paper compares with other algorithms as the dimensionality increases.

We conducted the Friedman test to determine sample chi-squareness, with one factor tested on the data and the other factor used to differentiate between groups of zones. The computational procedure was selected after 25 independent runs of each function at each algorithm size, allowing for the computation of three different dimensions.

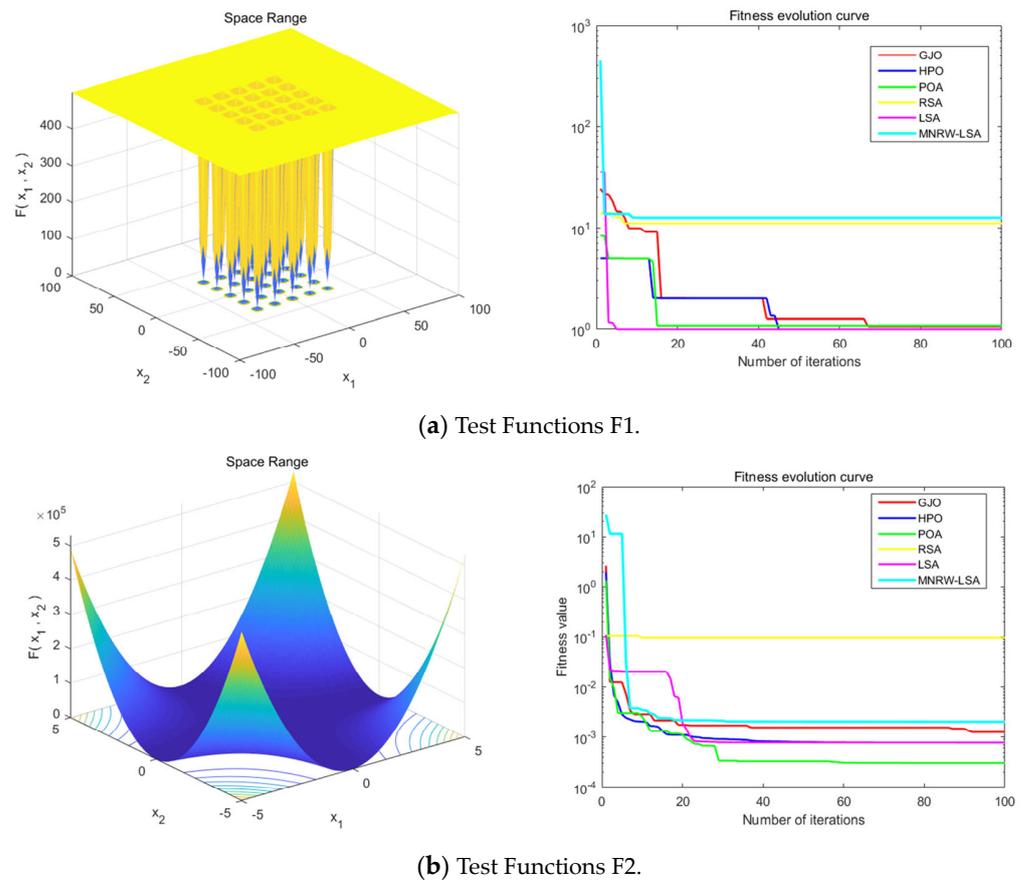
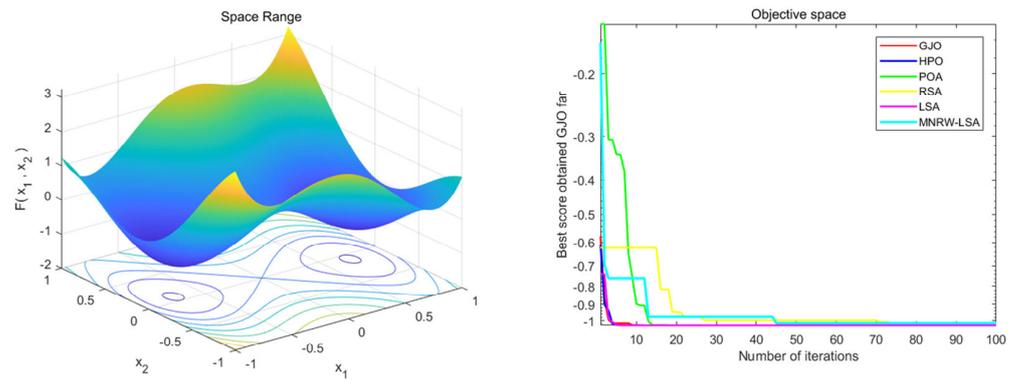
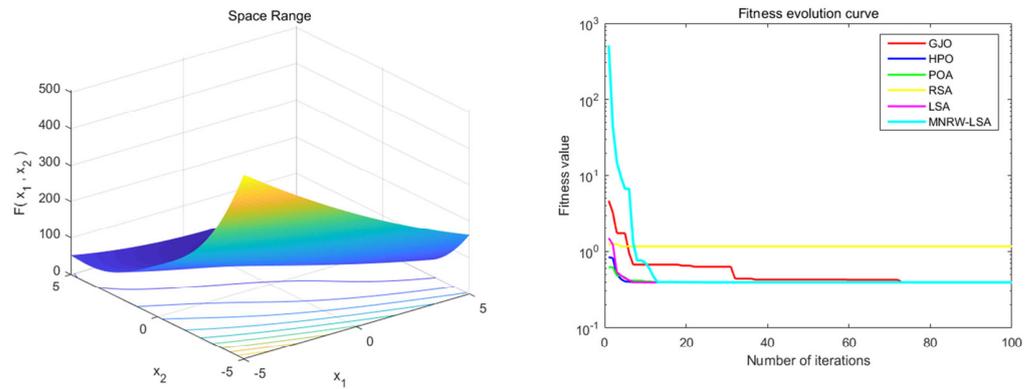


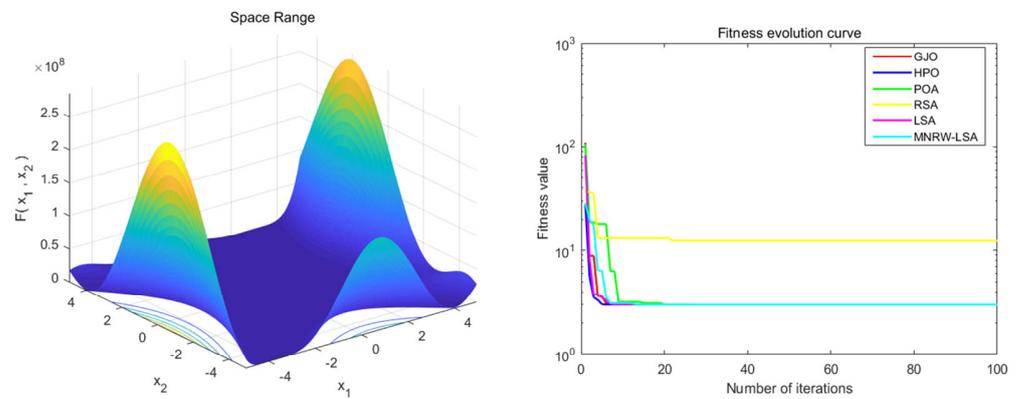
Figure 8. Cont.



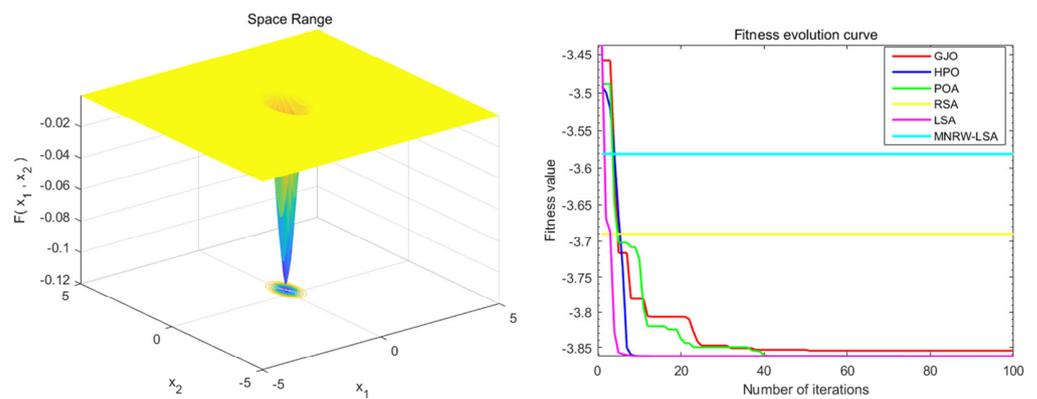
(c) Test Functions F3.



(d) Test Functions F4.

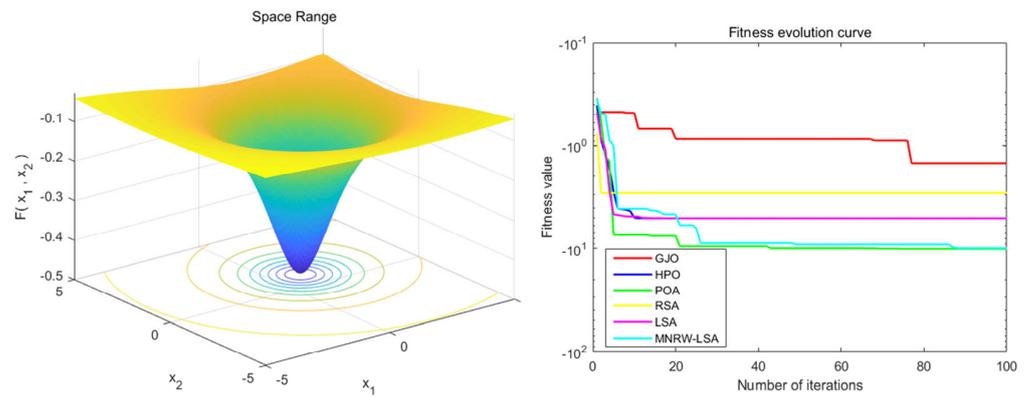


(e) Test Functions F5.

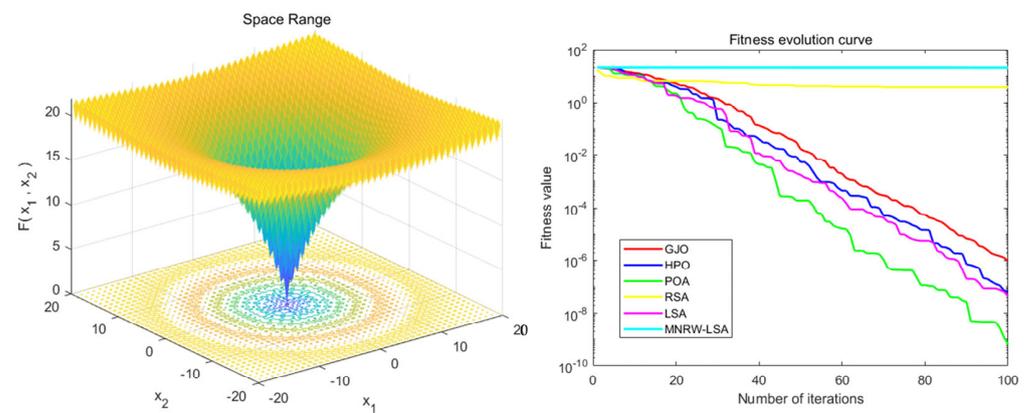


(f) Test Functions F6.

Figure 8. Cont.



(g) Test Functions F7.



(h) Test Functions F8.

Figure 8. (a–c) Dim = 30, comparison of the average curves of GJO, HPO, POA, RSA, LSA, and MNRW-LSA algorithms. (d–f) Dim = 30, comparison of the average curves of GJO, HPO, POA, RSA, LSA, and MNRW-LSA algorithms. (g,h) Dim = 30, comparison of the average curves of GJO, HPO, POA, RSA, LSA, and MNRW-LSA algorithms.

Table 4 summarizes the final results obtained by the algorithm in 50, 100, and 200 dimensions as a function of the average final error of the function described as less than 10^{-10} . Table 5 shows the results of the Friedman test, Table 6 shows the calculated knots of the average sequence.

Table 4. Number of functions successfully solved by each algorithm in three dimensions (average error below 8×10^3).

	GJO	HPO	POA	RSA	LSA	MNRW-LSA
D = 50	4	3	4	6	6	7
D = 100	4	3	4	4	4	5
D = 200	3	2	4	3	4	4

Based on the p -values in Table 5, it can be inferred that MNRW-LSA is significantly different from the other algorithms at a confidence level because all p -values (0.0346, 0.0318, and 0.0271) are less than 0.05. However, further testing is needed to determine which two algorithms are significantly different from each other.

Table 5. Friedman’s test for the results in the three dimensions.

	D = 50	D = 100	D = 200
GJO	4.126	4.102	4.071
HPO	3.324	3.322	4.286
POA	4.314	4.267	4.251
RSA	3.585	3.539	3.482
LSA	3.114	3.047	2.982
MNRW-LSA	2.846	2.844	2.786
<i>p</i>	0.0346	0.0318	0.0271

Table 6. Average sequential value difference between algorithms in the three dimensions.

	vs. GJO	vs. HPO	vs. POA	vs. RSA	vs. LSA
D = 50	2.921	2.842	2.718	2.734	2.698
D = 100	2.894	2.733	2.622	2.722	2.681
D = 200	2.812	2.651	2.549	2.708	2.673

Subsequently, we further distinguish the algorithms using the Nemenyi follow-up test to calculate the critical value domain of the mean ordinal value difference using Equation (45).

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}, \tag{45}$$

We bring in $\alpha = 0.05$ and $k = 4$ to calculate $q_\alpha = 2.728$. If the difference between the average sequential values of the algorithms exceeds the critical value domain, the hypothesis that the two algorithms perform equally is rejected with the corresponding confidence level.

Table 6 demonstrates that there is not a significant difference between the algorithms at low dimensions, and even the disparity between MNRW-LSA and LSA algorithms is minimal under these conditions. However, as the dimensionality increases, the contrast becomes more apparent, and the MNRW-LSA algorithm displays a significant advantage over other algorithms.

5.3. Validity Test of GP-RRT Algorithm

Four UAVs, a common starting point, and four target points are set in this paper and simulated using Matlab R2017a to make the UAV path planning typical and complete the optimization calculation of the index of coverage in the objective function.

To verify the performance of the improved RRT algorithm, first, we used a three-dimensional sphere to conduct a simulation test. We compared RRT, RRT-connet, and GP-RRT. The test process is shown in Figure 9, and the data parameters are shown in Table 7.

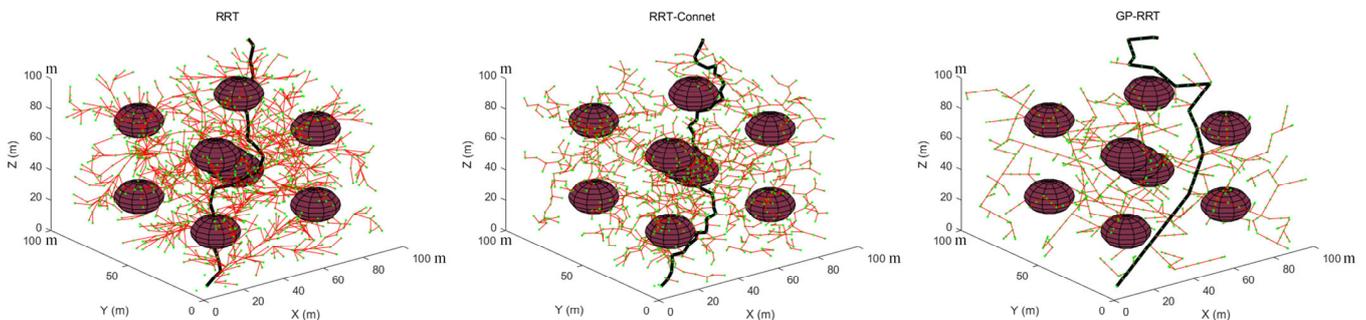


Figure 9. GP-RRT algorithm improvement test comparison.

Where the black lines are the paths finally found by the three algorithms, the red lines are the branches of the expanded tree, and the green dots are randomly sampled points in the three-dimensional space. It can be seen from Table 7 that the number of effective nodes of the GP-RRT algorithm proposed in this paper is significantly higher than that of the other two algorithms, which is an increase of about 63.17%. The average path length was shortened from 234.79 m to 183.93 m, a reduction of about 21.66%. The average running time is relatively higher than RRT-connet, but compared with RRT, the time is shortened by nearly 48%. The above results prove that the GP-RRT algorithm makes the path more directional, greatly reduces the number of unnecessary nodes, and effectively improves performance.

Table 7. GP-RRT algorithm improvement test comparison data (a unit is one meter in length).

Name/Parameter	Average Running Time	Average Number of Sampling Points	Average Number of Effective Nodes	Average Path Length	Average Percentage of Effective Nodes
RRT	About 25 s	2516.81	84.16	234.79 m	3.34
RRT-connet	About 7 s	1247.64	46.61	207.35 m	3.74
GP—RRT	About 13 s	761.91	41.51	184.93 m	5.45

5.4. UAV Path Planning Optimization Results

We first use the Dijkstra algorithm to preprocess the three-dimensional topographic map and obtain the preliminary topographic path distribution, as shown in Figure 10.

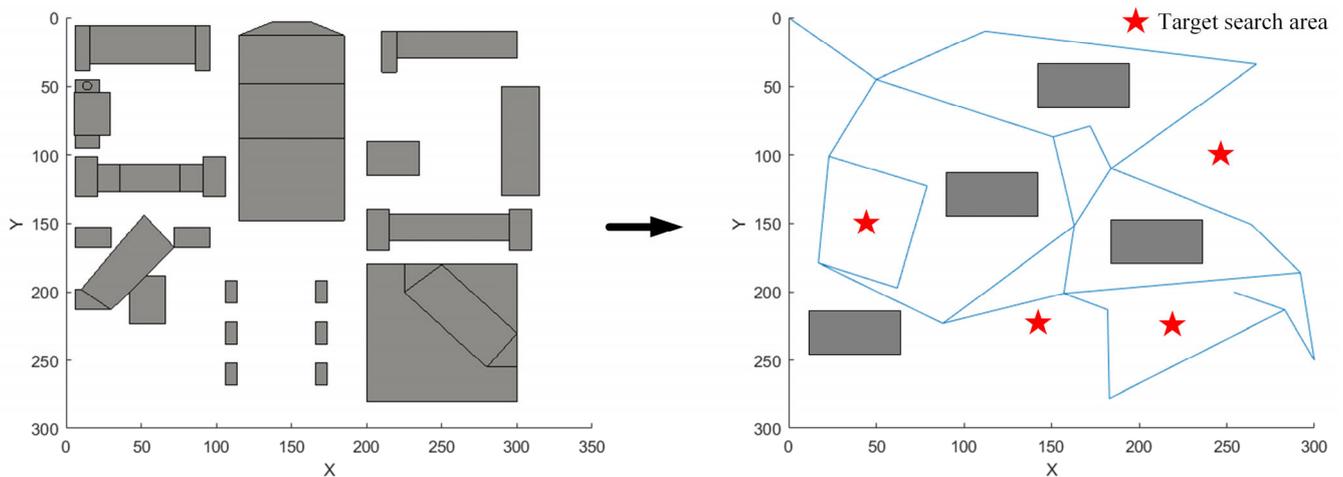


Figure 10. Basic path planning area.

After obtaining the path segmentation, we use the GP-RRT algorithm to obtain the initial three-dimensional planning path and use the PSO, SA, LSA, and MNRW-LSA algorithms to redistribute and optimize the coordinate values of each location point of each UAV. When flying close to a building, control the altitude change to $[-1, 1]$. Thus, the irregular variation loss in height is saved. The pitch angle changes of the four UAVs at each node are shown in Figure 11b.

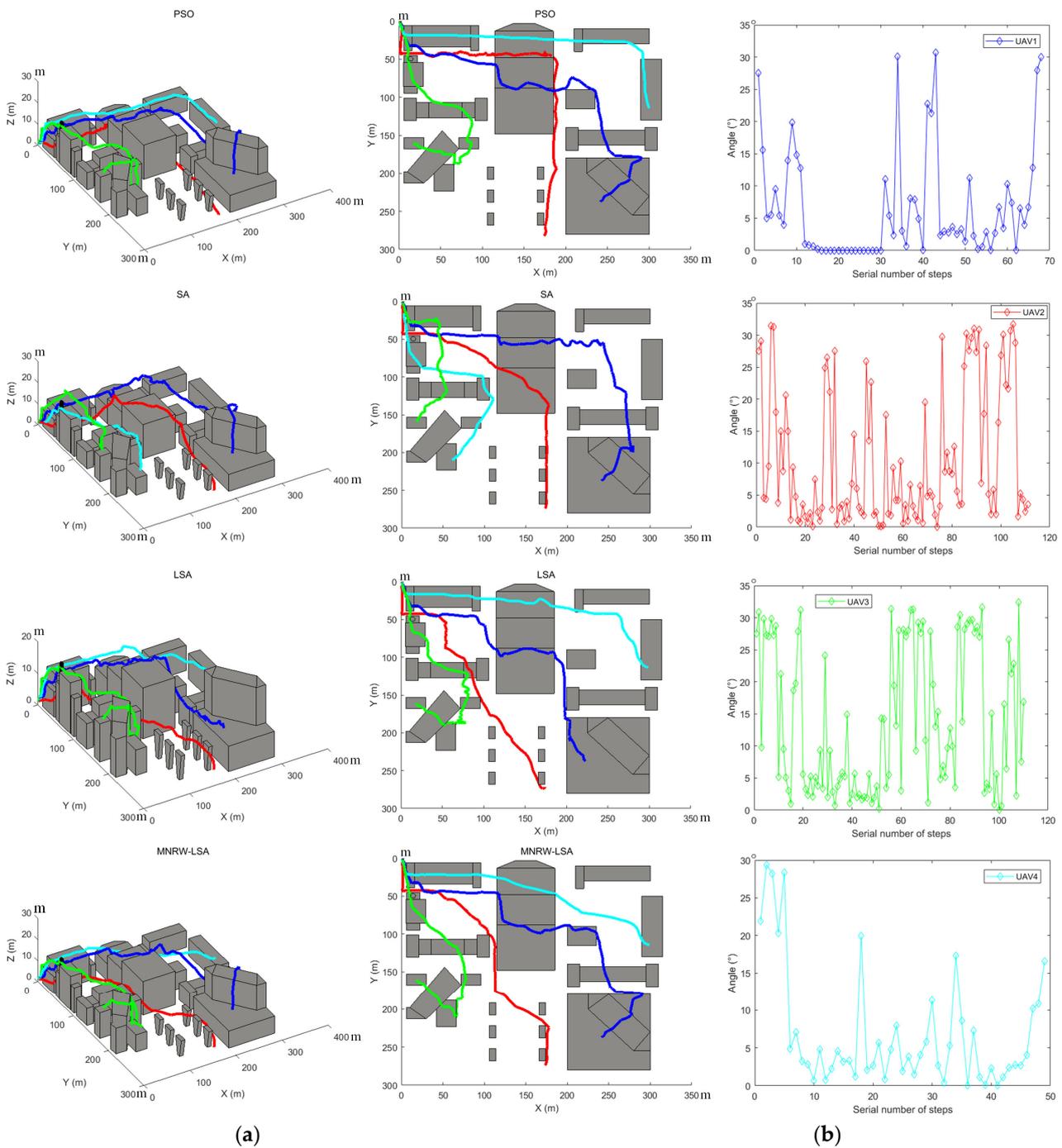
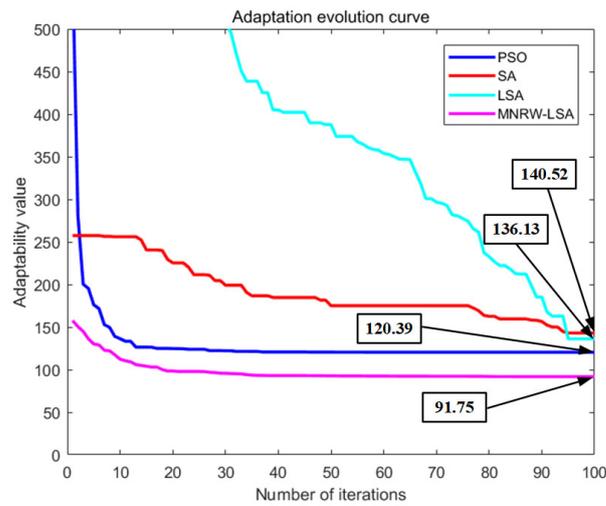
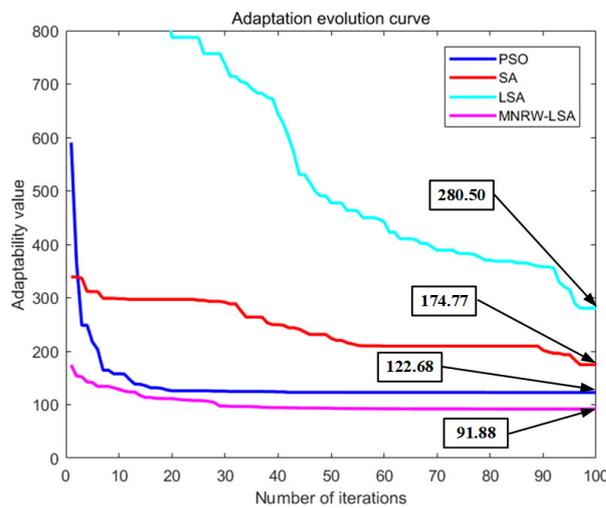


Figure 11. (a) Four algorithms to optimize the path results and (b) changes in pitch angle of four UAVs ($^{\circ}$).

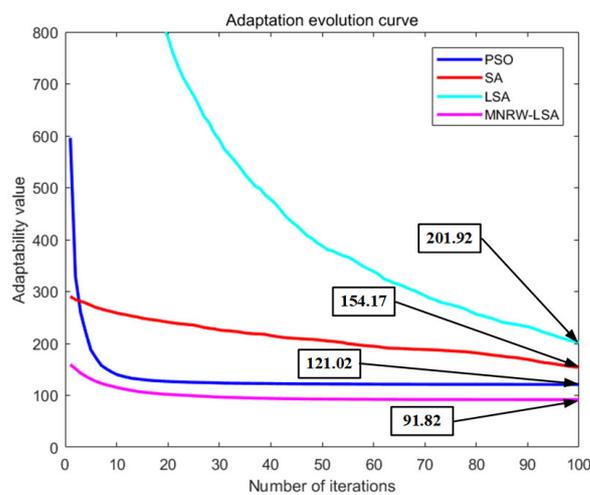
The results presented in Figure 12 demonstrate that the MNRW-LSA algorithm outperforms other algorithms in terms of optimal solution value under the same number of iterations and population size. Specifically, the optimal solution value of MNRW-LSA is 91.75, which is significantly better than the optimal solution values of standard PSO (120.39), standard SA (136.13), and standard LSA (140.52). These findings suggest that the MNRW-LSA algorithm is not only faster but also more accurate in terms of convergence speed and accuracy compared with the other algorithms. The energy loss of each node of the four UAVs is shown in Figure 13.



(a) Curve of the best results of 30 runs.



(b) Curve of the worst results of 30 runs.



(c) Curve of the average results of 30 runs.

Figure 12. (a,b) Thirty runs of best, worst, and average results. (c) Thirty runs of best, worst, and average results.

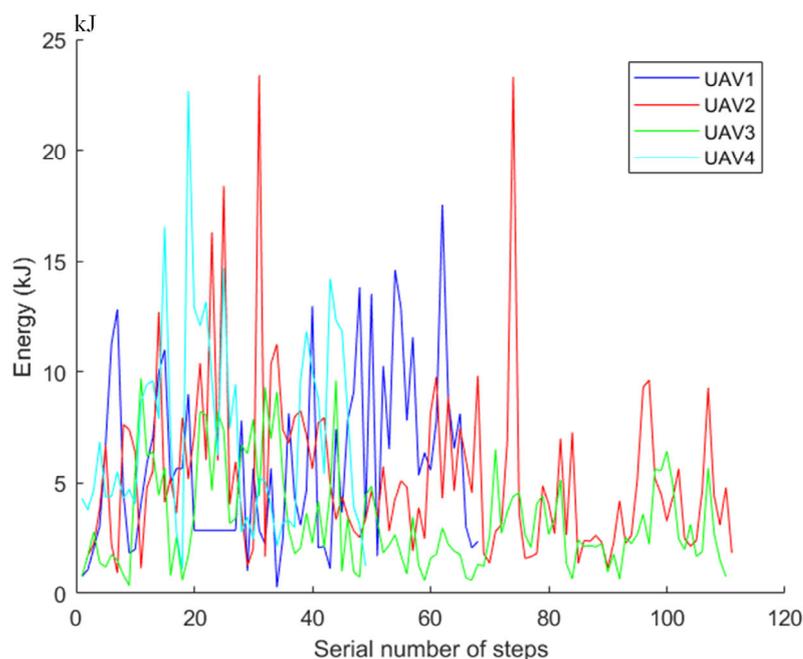


Figure 13. Energy changes curve.

In algorithm analysis, the total number of executions of the statement $T(n)$ is a function of the problem size n , and then the change of $T(n)$ when n is analyzed, and the order of magnitude of $T(n)$ is determined. The time complexity of the algorithm, which is the time measure of the algorithm, is denoted as $T(n) = O(f(n))$. It means that with the increase of the problem size n , the length rate of the algorithm execution time is the same as the length rate of $f(n)$, which is called the asymptotic time complexity of the algorithm, referred to as time complexity, where $f(n)$ is some function of the problem gauge $O(1) < O(\log n) < O(n) < O(n \times \log n) < O(n^2) < O(n^3) < O(2^n) < O(n!) < O(n^n)$. According to the order in which the time complexity is arranged n , when n approaches infinity, only the terms of higher order are retained so that the time complexity results obtained by the analysis are more efficient and reliable. The time complexity of the MNRW-LSA algorithm is $T_2(n) = O((n + D + n^D)^2) = O(n^{2D})$, and the time complexity of the LSA algorithm is $T_2(n) = O((n + D + n^{D+r})^2) = O(n^{2D+2r})$, where D is the population dimension and r is the number of duplicate channels.

In the MNRW-LSA algorithm, the loss of a more efficient discharge channel is prevented due to the improved method, which makes it more efficient to find an efficient channel. At the same time, the random walk strategy enables the algorithm to avoid repeated path optimization, which improves the overall optimization efficiency of the algorithm. Therefore, the algorithm should have a shorter run time and higher optimization efficiency than the LSA algorithm.

According to the data in Table 8, in running the three algorithms 30 times, the results of the algorithm proposed in this paper are all the best from the perspective of any of the three indicators: the optimal solution, the worst solution, and the average value. The paper presents an improved version of the MNRW-LSA algorithm that shows higher convergence accuracy and faster speed in solving the problem of simultaneous urban patrol path optimization by multiple UAVs. The algorithm also exhibits strong robustness and uniformity, making it a promising approach for this task.

Table 8. Objective function solution results.

	PSO	SA	LSA	MNRW-LSA
Optimal solution	120.39	136.13	140.52	91.75
Worst solution	122.68	174.77	280.50	91.88
Average solution	121.02	154.17	201.92	91.82
Optimal path length	1802.49 m	1621.69 m	1653.53 m	1497.03 m
Optimal coverage	74.42%	82.37%	83.99%	86.20%
Average running time	6.0476 s	4.8236 s	8.4435 s	5.8985 s

At the same time, the optimal path length obtained by the MNRW-LSA algorithm is shortened by 305.46 m at most and 124.66 m at least. Compared with other algorithms, the optimal coverage rate has increased by 11.78% at most, reaching 86.20%, indicating that the path is well planned, and most of the building groups and other spaces can be observed.

However, the average running time of the MNRW-LSA algorithm is longer than that of the PSO and SA algorithms. After analyzing the process and principle of each algorithm, we have identified that the MNRW-LSA algorithm takes longer to run due to the inclusion of the cross-fetching process. This process is repeated in each iteration to ensure the accuracy of the results, resulting in a running time that is approximately 1 s longer than other algorithms. However, compared with the basic LSA algorithm, the number of times the channel time is reset to zero is reduced due to the multi-level nesting process, which significantly speeds up the running time.

6. Conclusions and Future Research

This paper explores the patrolling of multi-UAVs in urban environments and the optimization of their trajectories. A three-dimensional topographic map was created for certain building areas in Beijing, China, to support this research. The study takes into account the UAV's three-dimensional flight constraints, performance constraints, and terrain constraints to establish a path cost model for multiple UAVs to patrol together.

To optimize the UAV trajectory cost in the model, we have enhanced the classical RRT algorithm by incorporating a greedy strategy inspired by optimization algorithms. This has led to the development of the GP-RRT algorithm, which effectively generates an initial path for path arrangement. Our focus is on achieving both extensive patrol coverage and a short path during UAV path planning. The patrol direction of the UAV is partitioned into larger areas so that there is no overlapping of UAV patrol areas as far as possible.

This paper proposes the use of a multi-layer nesting strategy and random tour strategy to enhance the performance of the lightning search algorithm. The resulting algorithm, MNRW-LSA, is then compared to other algorithms, such as PSO, SA, and LSA. The study shows that MNRW-LSA outperforms these other algorithms. The results show that the MNRW-LSA algorithm optimizes the path length to 1497.03 m and achieves a coverage rate of 86.20%. Its convergence speed and convergence accuracy are significantly higher than other algorithms. The algorithm effectively optimizes the UAV's trajectory in urban patrols and proves the correctness of the model. It has great advantages in solving the Multi-UAV trajectory planning problem.

The starting point of a UAV changes based on real-time data obtained from the road detection system, such as navigation. This allows a UAV to adapt to changes in vehicle and population density, completing its work in a real-time environment. The contributions of the paper are summarized as follows.

1. This manuscript introduces the concept of UAV urban patrol, which addresses the need for efficient surveillance and monitoring in urban environments. Furthermore, a reality-based model of the urban environment is developed, providing a realistic representation for evaluating the proposed solutions.
2. In order to address the challenges of UAV urban patrol effectively, the mathematical model has been expanded to consider various factors. These include the impact cost of UAV operations, flight energy consumption cost, and mission execution rate.

Moreover, the model takes into account the constraints imposed by the UAV's flight range and range limitation, resulting in a comprehensive approach to optimizing UAV patrol strategies.

3. A novel approach called the multi-level nesting and random walk strategy (MNRW-LSA) has been developed to enhance the performance of the lightning search algorithm (LSA) and achieve better resource allocation optimization. The MNRW-LSA algorithm incorporates multiple levels of nesting and employs a random walk strategy to improve search efficiency and accuracy, enabling more effective allocation of resources for UAV urban patrol.
4. The reliability and integrity of the MNRW-LSA model are validated through simulation experiments. These experiments demonstrate the effectiveness of the proposed algorithm in comparison to other existing algorithms commonly used in UAV urban patrol scenarios. The comparison provides valuable insights into the superior performance and benefits of the MNRW-LSA approach.

This paper provides a new improvement idea for UAV path planning, which can effectively solve the related planning problems in static path planning. However, there are still limitations in the current research process. For example, the interference of weather factors during UAV flight and the changes in vehicle and population density in the real-time environment are not considered. In future research, more improvement strategies should be tried to be introduced, which will better improve the algorithm performance. At the same time, we will consider applying machine learning methods to learn and optimize the search strategies of heuristic algorithms to improve the efficiency of solving and the quality of solutions, thereby addressing more complex task environments.

Author Contributions: Conceptualization, Y.H. and H.X.; methodology, Y.H.; software, H.X.; validation, Y.H., H.X. and M.Z.; formal analysis, Y.H.; investigation, N.P.; resources, N.P.; data curation, Z.W.; writing—original draft preparation, Y.H.; writing—review and editing, Y.H. and H.X.; visualization, H.X.; supervision, N.P.; project administration, Y.H.; funding acquisition, N.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China Southern Power Grid (grant No. YNKJXM20220174) and the APC was funded by Nan Pan.

Data Availability Statement: No new data were created.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Tian, G.; Liu, J.; Zhao, H.; Yang, W. Small object detection via dual inspection mechanism for UAV visual images. *Appl. Intell.* **2021**, *52*, 4244–4257. [[CrossRef](#)]
2. Qadir, Z.; Ullah, F.; Munawar, H.S.; Al-Turjman, F. Addressing disasters in smart cities through UAVs path planning and 5G communications: A systematic review. *Comput. Commun.* **2021**, *168*, 114–135. [[CrossRef](#)]
3. Mukhamediev, R.I.; Symagulov, A.; Yergaliyeva, B.; Madiyev, Y.; Kaltayev, A.; Zhumabekova, G. Review of some applications of unmanned aerial vehicles technology in the resource-rich country. *Appl. Sci.* **2021**, *11*, 10171. [[CrossRef](#)]
4. Li, Y.; Liu, M.; Jiang, D. Application of Unmanned Aerial Vehicles in Logistics: A Literature Review. *Sustainability* **2022**, *14*, 14473. [[CrossRef](#)]
5. Liu, H.; Chen, Q.; Pan, N.; Sun, Y.; An, Y.; Pan, D. UAV Stocktaking Task-Planning for Industrial Warehouses Based on the Improved Hybrid Differential Evolution Algorithm. *IEEE Trans. Ind. Inform.* **2021**, *18*, 582–591. [[CrossRef](#)]
6. Liu, H.; Chen, Q.; Wang, J.; An, Y.; Pan, N.; Pan, D. Study on UAV Parallel Planning System for Transmission Line Project Acceptance Under the Background of Industry 5.0. *IEEE Trans. Ind. Inform.* **2022**, *18*, 5537–5546. [[CrossRef](#)]
7. Arokiasami, W.A.; Vadakkepat, P.; Tan, K.C.; Srinivasan, D. Interoperable multi-agent framework for unmanned aerial/ground vehicles: Towards robot autonomy. *Complex Intell. Syst.* **2016**, *2*, 45–59. [[CrossRef](#)]
8. Yao, P.; Wang, H.; Su, Z. UAV feasible path planning based on disturbed fluid and trajectory propagation. *Chin. J. Aeronaut.* **2015**, *28*, 1163–1177. [[CrossRef](#)]
9. Xia, Q.; Liu, S.; Guo, M.; Wang, H.; Zhou, Q.; Zhang, X. Multi-UAV trajectory planning using gradient-based sequence minimal optimization. *Robot. Auton. Syst.* **2021**, *137*, 103728. [[CrossRef](#)]
10. Gao, S.; Ai, J.L.; Wang, Z.H. Mixed population RRT algorithm for UAV path planning. *Syst. Eng. Electron.* **2020**, *42*, 101–107.

11. Wang, T.; Huang, P.F.; Dong, G.Q. Cooperation path planning of multi-UAV in road-network continuous monitoring. *Acta Aeronaut. Astronaut. Sin.* **2020**, *41*, 61–70. [[CrossRef](#)]
12. Zhang, H.H.; Gan, X.H.; Li, S.F.; Feng, Z.; Jin, Y. UAV route planning considering regional risk assessment under complex low altitude environment. *Chin. J. Sci. Instrum.* **2021**, *42*, 257–266.
13. Na, Z.; Zhang, M.C.; Low, K.H. 3D path planning and real-time collision resolution of multirotor drone operations in complex urban low-altitude airspace. *Transp. Res. Part C Emerg. Technol.* **2021**, *129*, 103–123.
14. Muñoz, J.; López, B.; Quevedo, F.; Monje, C.A.; Garrido, S.; Moreno, L.E. Multi UAV Coverage Path Planning in Urban Environments. *Sensors* **2021**, *21*, 7365. [[CrossRef](#)] [[PubMed](#)]
15. Zhang, Q.R.; Wei, R.X.; He, R.K.; Ru, C.J.; Zhou, K. Path planning for unmanned aerial vehicle in urban space crowded with irregular obstacles. *Control Theory Appl.* **2015**, *32*, 1407–1413.
16. Khan, S.I.; Qadir, Z.; Munawar, H.S.; Nayak, S.R.; Budati, A.K.; Verma, K.; Prakash, D. UAVs path planning architecture for effective medical emergency response in future networks. *Phys. Commun.* **2021**, *47*, 101337. [[CrossRef](#)]
17. Causa, F.; Fasano, G. Multiple UAVs trajectory generation and waypoint assignment in urban environment based on DOP maps. *Aerosp. Sci. Technol.* **2021**, *110*, 106507. [[CrossRef](#)]
18. Dai, R.; Fotedar, S.; Radmanesh, M.; Kumar, M. Quality-aware UAV coverage and path planning in geometrically complex environments. *Ad Hoc Netw.* **2018**, *73*, 95–105. [[CrossRef](#)]
19. Yuan, J.; Li, S. UAV 3D Path Planning and Obstacle Avoidance Method. *Inf. Control* **2021**, *50*, 95–101.
20. He, W.; Qi, X.; Liu, L. A novel hybrid particle swarm optimization for multi-UAV cooperate path planning. *Appl. Intell.* **2021**, *51*, 7350–7364. [[CrossRef](#)]
21. Maini, P.; Sundar, K.; Singh, M.; Rathinam, S.; Sujit, P.B. Cooperative Aerial–Ground Vehicle Route Planning with Fuel Con-straints for Coverage Applications. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 3016–3028. [[CrossRef](#)]
22. Bolourian, N.; Hammad, A. LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection. *Autom. Constr.* **2021**, *117*, 103250. [[CrossRef](#)]
23. Li, K.; Ge, F.; Han, Y.; Wang, Y.; Xu, W. Path planning of multiple UAVs with online changing tasks by an ORPFOA algorithm. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103807. [[CrossRef](#)]
24. Huang, C.Q.; Zhao, K.X.; Han, B.J.; Wei, Z.L. Maneuvering Decision-making Method of UAV Based on Approximate Dy-namic Programming. *J. Electron. Inf. Technol.* **2018**, *40*, 2447–2452.
25. Belkadi, A.; Abaunza, H.; Ciarletta, L.; Castillo, P.; Theilliol, D. Design and Implementation of Distributed Path Planning Algorithm for a Fleet of UAVs. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 2647–2657.
26. Goez, G.D.; Velasquez, R.; Botero, J.S. UAV route planning optimization using PSO implemented on microcontrollers. *IEEE Lat. Am. Trans.* **2016**, *14*, 1705–1710. [[CrossRef](#)]
27. Liu, Q.; Zhang, Y.; Li, M.; Zhang, Z.; Cao, N.; Shang, J. Multi-UAV Path Planning Based on Fusion of Sparrow Search Algorithm and Improved Bioinspired Neural Network. *IEEE Access* **2021**, *9*, 124670–124681. [[CrossRef](#)]
28. Sun, J.; Zhang, Q.F.; Hui, B.; Chang, Z.; Xu, Z.M. UAV Object Tracking for Battlefield Reconnaissance Based on a Chaot-ic-mutated Bat Algorithm. *Inf. Control* **2018**, *47*, 140–148.
29. Duan, H.; Zhao, J.; Deng, Y.; Shi, Y.; Ding, X. Dynamic Discrete Pigeon-Inspired Optimization for Multi-UAV Cooperative Search-Attack Mission Planning. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *57*, 706–720. [[CrossRef](#)]
30. Jain, G.; Yadav, G.; Prakash, D.; Shukla, A.; Tiwari, R. MVO-based path planning scheme with coordination of UAVs in 3-D environment. *J. Comput. Sci.* **2019**, *37*, 101016. [[CrossRef](#)]
31. Abualigah, L.; Elaziz, M.A.; Hussien, A.G.; Alsalibi, B.; Jalali, S.M.J.; Gandomi, A.H. Lightning search algorithm: A comprehensive survey. *Appl. Intell.* **2020**, *51*, 2353–2376. [[CrossRef](#)] [[PubMed](#)]
32. Nc, A.; Mma, B. Golden Jackal Optimization: A Novel Nature-Inspired Optimizer for Engineering Applications. *Expert Syst. Appl.* **2022**, *198*, 116924.
33. Naruei, I.; Keynia, F.; Sabbagh Molahosseini, A. Hunter-prey optimization: Algorithm and applications. *Soft Comput.* **2022**, *26*, 1279–1314. [[CrossRef](#)]
34. Trojovský, P.; Dehghani, M. Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications. *Sensors* **2022**, *22*, 855. [[CrossRef](#)]
35. Abualiga, H.L.; Elaziz, M.A.; Sumari, P.; Geem, Z.W.; Gandomi, A.H. Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Syst. Appl.* **2021**, *191*, 116158. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.