aerospace

MDPI

Article

Adaptive Strategy for the Path Planning of Fixed-Wing UAV Swarms in Complex Mountain Terrain via Reinforcement Learning

Lei Lv [1], Wei Jia [2,3], Ruofei He [3] and Wei Sun [1,*]

[1] School of Space Science and Technology, Xidian University, Xi'an 710118, China; 21131110572@stu.xidian.edu.cn
[2] Xi'an ASN Technology Group Co., Ltd., Xi'an 710065, China; jiawei@nwpu.edu.cn
[3] 365th Research Institute, Northwestern Polytechnical University, Xi'an 710072, China; heruofei@nwpu.edu.cn
* Correspondence: wsun@xidian.edu.cn

Abstract

Cooperative path planning for multiple Unmanned Aerial Vehicles (UAVs) within complex mountainous terrain presents a unique challenge, characterized by a high-dimensional search space fraught with numerous local optima. Conventional metaheuristic algorithms often fail in such deceptive landscapes due to premature convergence stemming from a static balance between exploration and exploitation. To overcome the aforementioned limitations, this paper develops the Reinforcement Learning-guided Hybrid Sparrow Search Algorithm (RLHSSA), an optimization framework specifically engineered for robust navigation in complex topographies. The core innovation of RLHSSA lies in its two-level architecture. At a lower level, a purpose-built operator suite provides specialized tools essential for mountain environments: robust exploration strategies, including Levy Flight, to escape the abundant local optima, and an Elite-SSA for the high-precision exploitation needed to refine paths within narrow corridors. At a higher level, a reinforcement learning agent intelligently selects the most suitable operator to adapt the search strategy to the terrain's complexity in real-time. This adaptive scheduling mechanism is the key to achieving a superior exploration–exploitation balance, enabling the algorithm to effectively navigate the intricate problem landscape. Extensive simulations within challenging mountainous environments demonstrate that RLHSSA consistently outperforms state-of-the-art algorithms in solution quality and stability, validating its practical potential for high-stakes multi-UAV mission planning.

Keywords: multi-UAV path planning; sparrow search; reinforcement learning; optimization algorithm

check for updates

Academic Editor: Carlo E. D. Riboldi

Received: 9 October 2025
Revised: 17 November 2025
Accepted: 18 November 2025
Published: 19 November 2025

Citation: Lv, L.; Jia, W.; He, R.; Sun, W. Adaptive Strategy for the Path Planning of Fixed-Wing UAV Swarms in Complex Mountain Terrain via Reinforcement Learning. *Aerospace* **2025**, *12*, 1025. https://doi.org/10.3390/aerospace12111025

Copyright: © 2025 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

Unmanned aerial vehicles (UAVs) have achieved remarkable advancements and are now broadly employed across various fields, including surveillance, agriculture, and environmental monitoring [1–3]. Compared to traditional manned-aircraft platforms, UAVs offer unparalleled flexibility and can collect high spatial and temporal resolution data at lower cost by operating at low altitudes [4]. However, deploying UAVs for missions in complex and challenging environments presents significant difficulties [5]. Mountainous areas, for instance, are characterized by intricate terrain, steep slopes, and varying altitudes, which pose substantial obstacles to safe and efficient flight [6]. Successful mission execution in complex, hazardous environments depends on the UAV's autonomous and safe navigation [7,8].

For many UAV tasks, utilizing a single UAV may be inefficient, especially when covering large areas [9,10]. In such scenarios, multi-UAV systems offer distinct advantages for rapid data acquisition, including enhanced coverage, reduced mission time, and increased robustness through cooperation [11–13]. Nevertheless, navigating a swarm of UAVs through complex 3D terrain—while ensuring collision avoidance with terrain, obstacles, and each other, and simultaneously considering threats and other operational constraints—presents a formidable navigation challenge [14–16]. Therefore, developing effective cooperative path planning strategies is a critical requirement for advancing UAV capabilities in challenging real-world applications.

The challenge of multi-UAV cooperative path planning within intricate environments remains a focal point of research. Solutions can often be broadly categorized into traditional methods, such as graph-based search algorithms (e.g., A* [17], Dijkstra [18]) and sampling-based methods (e.g., rapidly exploring random trees [19]), alongside optimization-based techniques including mathematical programming and metaheuristic algorithms [20–22]. While traditional methods are effective in well-defined or lower-dimensional spaces, they often face scalability issues and computational challenges when applied to high-dimensional, continuous, and complex 3D environments with numerous obstacles and constraints, which are typical of realistic mountainous terrains. Optimization-based methods, particularly metaheuristic algorithms including genetic algorithms (GAs) [23], particle swarm optimization (PSO) [24], and grey wolf optimizer (GWO) [25], have shown promise in handling complex search spaces and multiple objectives. For example, Zhang et al. [26] enhanced the differential evolution (DE) algorithm through the integration of multiple strategies, proposing a novel variant, namely MSIDE. This method was specifically tailored for UAV 3D path planning in challenging mountainous scenarios, aiming to provide effective solutions for this intricate problem. Meng et al. [27] developed an evolutionary state estimation-based multi-strategy jellyfish search (ESE-MSJS) algorithm to address path planning for multiple UAVs in intricate environments. Their approach integrates a switching framework based on evolutionary state estimation and employs various learning strategies to enhance search efficiency and adapt to complex constraints.

Furthermore, reinforcement learning (RL) provides an effective methodology where an agent acquires optimal actions or strategies by interacting with its surroundings, aiming to maximize cumulative rewards [28]. RL has gained significant attention in various fields, including robotics, control, and intelligent decision making [29]. Researchers have explored integrating RL with optimization algorithms to enhance their search capabilities or adapt their parameters during the optimization process [30,31].

However, directly applying standard metaheuristic algorithms to this multi-UAV routing task reveals a critical vulnerability when faced with the unique topography of mountainous environments. Such landscapes create a deceptive search space characterized by numerous local optima and narrow, winding feasible corridors. A key limitation of conventional methods is their reliance on a static or overly simplistic adaptive balance between exploration and exploitation [32,33]. This inflexibility renders them prone to premature convergence within a suboptimal valley or inefficiently searching after a viable corridor has been found. These challenges highlight the urgent need for a more intelligent optimization paradigm that can dynamically adapt its search behavior to the specific topographical complexities it encounters.

To conquer these topographical challenges, we introduce a novel optimization framework: the Reinforcement Learning-guided Hybrid Sparrow Search Algorithm (RLHSSA). The primary innovation of RLHSSA is to replace the fixed search strategy of the standard Sparrow Search Algorithm (SSA) [34] with a learning-based, two-level architecture. Specifically, we enhance the SSA foundation by integrating a purpose-built suite of operators;

powerful exploration strategies, including Adaptive Differential Perturbation and Levy Flight, are incorporated to provide the long-jump capability necessary to escape the numerous local optima inherent in mountain landscapes. Simultaneously, an Elite-SSA operator provides the high-precision exploitation required to refine trajectories within narrow, viable corridors. The entire process is governed by an RL-based scheduling mechanism.

More importantly, the RL-based strategy scheduling mechanism allows the algorithm to intelligently select the most suitable operator throughout the optimization process based on feedback from the search environment. This adaptive strategy selection is expected to enable RLHSSA to better balance exploration and exploitation, adapt to different search stages, and effectively navigate the complex, high-dimensional search space associated with multi-UAV cooperative path planning in challenging mountainous and threatened environments. In summary, this paper makes the following key contributions:

(1) A novel Reinforcement Learning-guided Hybrid Sparrow Search Algorithm (RLHSSA) is proposed. This algorithm uniquely combines a base metaheuristic (SSA) with a suite of specialized operators and an adaptive RL framework to address complex multi-UAV path planning challenges, offering a more robust and efficient solution than single-strategy or non-adaptive approaches.

(2) Three distinct optimization operators—the Elite-SSA, the Adaptive Differential Perturbation strategy, and the Adaptive Levy Flight strategy—are designed and integrated into the framework. This integration provides a rich set of search behaviors, with Elite-SSA focusing on refined exploitation while the other two operators enhance search diversity and global exploration, thereby mitigating the common issue of premature convergence.

(3) A novel reinforcement learning-based mechanism is developed for the dynamic scheduling of distinct optimization operators within a hybrid metaheuristic. Unlike fixed-strategy or simpler adaptive methods, this mechanism enables the algorithm to intelligently select the most appropriate search behavior for the current optimization phase, leading to a superior exploration–exploitation balance and adaptability.

(4) Comprehensive simulation experiments and comparative studies were conducted to assess the effectiveness of the RLHSSA method against state-of-the-art methods in both threat-free and threatened environments. Our findings demonstrate RLHSSA's superior effectiveness concerning convergence speed, solution quality, and robustness, particularly in challenging mountainous scenarios.

The structure of this work is outlined as follows: Section 2 formulates the cooperative path planning problem for multiple UAVs, including the environment modeling and path representation. Section 3 focuses on constructing the problem model for mountainous terrain. Section 4 details the proposed RLHSSA algorithm. Section 5 discusses the simulation settings, experimental outcomes, and corresponding analysis across various scenarios. Finally, Section 6 presents conclusions and future works.

2. Problem Statement

Environmental modeling is a prerequisite for UAV path planning. For multi-UAV operations, the search space is defined by a mathematical representation of the operational environment. The undulating natural terrain in mountainous environments is modeled using an exponential function [24], whose mathematical formulation is provided below:

$$Z(x,y) = \sum_{j=1}^{m} h_j \exp\left[-\left(\frac{x-x_j}{x_{sj}}\right)^2 - \left(\frac{y-y_j}{y_{sj}}\right)^2\right]. \quad (1)$$

Here, (x_j, y_j) denotes the position of the j-th mountain peak, while h_j defines its elevation parameter. The values x_{sj} and y_{sj} control the gradient attenuation along the x- and y-axes, respectively, thereby influencing the steepness of the peak. Furthermore, m indicates the total count of mountain peaks. The environmental model constructed based on Equation (1) is shown in Figure 1a.

To ensure flight, UAVs must consider various potential threats during flight. This paper primarily considers two types of path threats: enemy threat sources and no-fly zones (NFZs). Enemy threat sources include radar, anti-aircraft guns, missiles, and electronic jamming, all of which have different detection ranges of interference strengths. NFZs are areas where flight is prohibited, designated by authorities. To simplify the calculation, we approximate enemy threat sources as hemispheres, with the radius representing the influence range of the threat. NFZs are modeled as cylinders, and a large penalty is imposed if the UAV enters the cylinder. In the model presented in this paper, we assume that the positions of all threat sources and NFZs, as well as their influence ranges, are known in advance. Figure 1b illustrates the 3D environmental model, which incorporates the previously described mountainous terrain and threat sources.

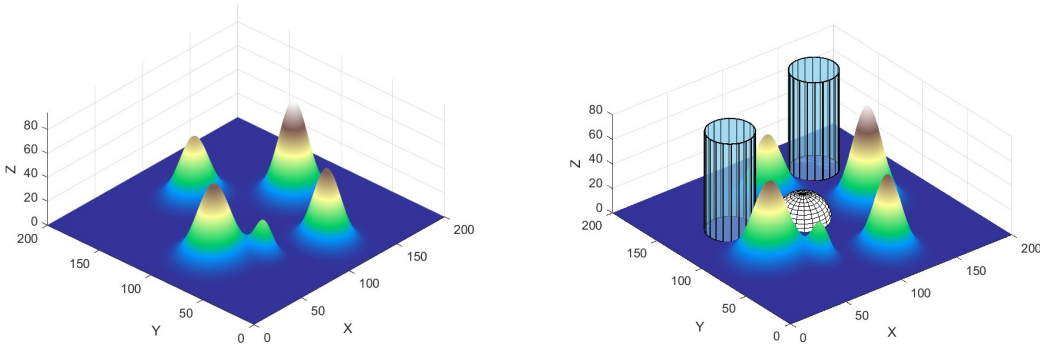


Figure 1. Environmental model. (**a**) Mountain obstacles. (**b**) Obstacle terrain.

Path Representation

In this system, N represents the total count of UAVs. For any UAV i in the formation $(i = 1, 2, \cdots, N)$, it needs to fly from the start point $S_i : (x_{i,s}, y_{i,s}, z_{i,s})$ to the end point $T_i : (x_{i,e}, y_{i,e}, z_{i,e})$. The flight path is composed of M waypoints, denoted as $p_{i,j} : (x_{i,j}, y_{i,j}, z_{i,j})$, $j = 1, 2, \cdots, M$. To simplify the modeling process, we define $p_{i,0} := S_i$, $p_{i,M+1} := T_i$. Thus, the route of UAV i is denoted by $P_i = \{p_{i,0}, p_{i,1}, \cdots, p_{i,M}, p_{i,M+1}\}$. Subsequently, the system's complete trajectory is given as $Path = \{P_1, P_2, \cdots, P_N\}$.

3. Multi-UAV Cooperative Path Planning

3.1. Flight Distance Cost

In this work, each UAV is modeled as a point-mass kinematic agent. For UAV i, the path segment formed by two consecutive waypoints $p_{i,j}$ and $p_{i,j+1}$ on its flight path is represented as $\overrightarrow{p_{i,j}p_{i,j+1}}$. For convenience, we define $\vec{L}_{i,j} := \overrightarrow{p_{i,j}p_{i,j+1}}$. The corresponding distance of this segment is computed as $\|\vec{L}_{i,j}\|$. The flight distance cost function F_1 is:

$$F_1(Path) = \sum_{i=1}^{N} \sum_{j=0}^{M} \|\vec{L}_{i,j}\|. \tag{2}$$

3.2. Flight Altitude Cost

Within the domain of 3D multi-UAV path planning, higher flight altitudes increase UAV susceptibility to enemy radar detection, whereas lower altitudes heighten the risk of

collision with the ground. Therefore, the flight altitude of UAVs must be set within a safe range, and altitude changes should be minimized during flight to ensure flight safety. The flight altitude cost function is given by:

$$Cost_{i,j} = \begin{cases} |z_{i,j+1} - z_{i,j}|, & \text{if } H_{\min} \leqslant (z_{i,j}, z_{i,j+1}) \leqslant H_{\max} \\ \infty, & \text{otherwise} \end{cases} \tag{3}$$

$$F_2(Path) = \sum_{i=1}^{N} \sum_{j=0}^{M} Cost_{i,j}, \tag{4}$$

where $H_{\min}$ and $H_{\max}$ correspond to lowest and highest allowable altitudes for the UAV. The values $z_{i,j}$ and $z_{i,j+1}$ are the z-axis positions for the UAV i at its j-th and $(j+1)$-th waypoints, respectively.

3.3. Path Stability Cost

As illustrated in Figure 2, when a UAV maneuvers between path nodes, large turning and climbing angles increase flight instability. Consequently, maintaining an appropriate distance between path nodes is vital for path rationality and stability; overly close nodes, for instance, would necessitate frequent attitude adjustments to meet multiple sequential waypoints quickly.

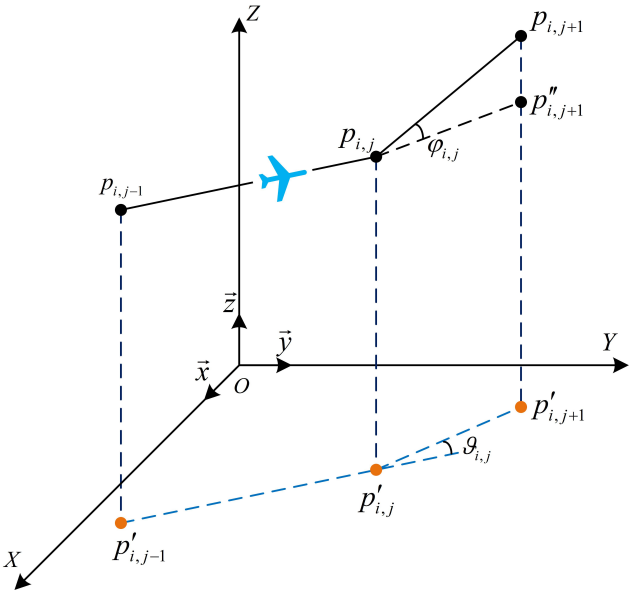


Figure 2. Diagram of UAV flight attitude and path segments.

(1) Turning angle and Climbing angle

The turning angle cost function of the UAV is calculated as follows:

$$\vartheta_{i,j} = \arccos\left(\frac{\vec{L}'_{i,j-1} \cdot \vec{L}'_{i,j}}{\|\vec{L}'_{i,j-1}\| \|\vec{L}'_{i,j}\|}\right) \tag{5}$$

$$\vec{L}'_{i,j-1} := \overrightarrow{p'_{i,j-1}p'_{i,j}}, \quad \vec{L}'_{i,j} := \overrightarrow{p'_{i,j}p'_{i,j+1}} \tag{6}$$

$$f_T(\vartheta_{i,j}) := \begin{cases} \vartheta_{i,j}, & \text{if } \vartheta_{i,j} > \vartheta_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $\vartheta_{i,j}$ signifies the turning angle of UAV i at its j-th waypoint, and $\vartheta_{\max}$ is the maximum turning angle of the UAV; $\vec{L}'_{i,j-1}$ and $\vec{L}'_{i,j}$ are the projection vectors of $\vec{L}_{i,j-1}$ and $\vec{L}_{i,j}$ onto the horizontal plane, respectively.

The climbing angle cost function of the UAV is [31]

$$\varphi_{i,j} = \arctan\left(\frac{z_{i,j+1} - z_{i,j}}{\|\vec{L}'_{i,j}\|}\right) \tag{8}$$

$$f_C(\varphi_{i,j}) := \begin{cases} |\varphi_{i,j} - \varphi_{i,j-1}|, & \text{if } |\varphi_{i,j} - \varphi_{i,j-1}| > \varphi_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $\varphi_{i,j}$ denotes the climb angle of UAV i at waypoint j, and $\varphi_{\max}$ specifies the upper bound of permissible climb angle change for the UAV.

The cost function concerning UAV angular variations is:

$$\text{Angle}_{i,j} := \sum_{i=1}^{N}\sum_{j=1}^{M} f_T(\vartheta_{i,j}) + \sum_{i=1}^{N}\sum_{j=1}^{M} f_C(\varphi_{i,j}) \tag{10}$$

The path stability cost function is given by:

$$F_3(Path) = \text{Angle}_{i,j}. \tag{11}$$

3.4. Path Threat Cost

(1) Enemy threat sources

Assume that there are K_1 enemy threat sources in the environment, denoted as $O_1, O_2, \cdots, O_r, \cdots, O_{K_1}$. Each threat source O_r is modeled as a hemispherical influence region with center C_r and detection radius d_r, i.e.,

$$O_r = \{p \in \mathbb{R}^3 | \|p - C_r\| \leqslant d_r\}. \tag{12}$$

For a given flight segment $L_{i,j}$, we say that it is affected by threat source O_r if and only if it passes through this influence region, i.e., $L_{i,j} \cap O_r \neq \emptyset$. As shown in Figure 3, to calculate the threat cost, five sub-nodes are extracted on each flight segment, located at positions 0.1, 0.3, ..., and 0.9 along the segment [35]. The enemy threat source cost is calculated as follows:

$$C^i_{j,r} := \begin{cases} G_r \cdot \left(\left|1 - \dfrac{D^i_{0.1,r,j}}{d_r}\right| + \left|1 - \dfrac{D^i_{0.3,r,j}}{d_r}\right| + \cdots \right. \\ \quad \left. + \left|1 - \dfrac{D^i_{0.9,r,j}}{d_r}\right|\right), \quad L_{i,j} \cap O_r \neq \emptyset \\ 0, \quad \text{otherwise} \end{cases} \tag{13}$$

$$f_E(\vec{L}_{i,j}) := \frac{\|\vec{L}_{i,j}\|}{10} \cdot \sum_{r=1}^{K_1} C^i_{j,r} \tag{14}$$

$$F_{threat}(Path) := \sum_{i=1}^{N}\sum_{j=0}^{M} f_E(\vec{L}_{i,j}) \tag{15}$$

The threat level of the r-th threat source is signified by G_r, and $D^i_{0.1,r,j}$ is defined as the distance from the center of threat source O_r to the subnode at the 0.1 position on the jth flight segment of UAV i.

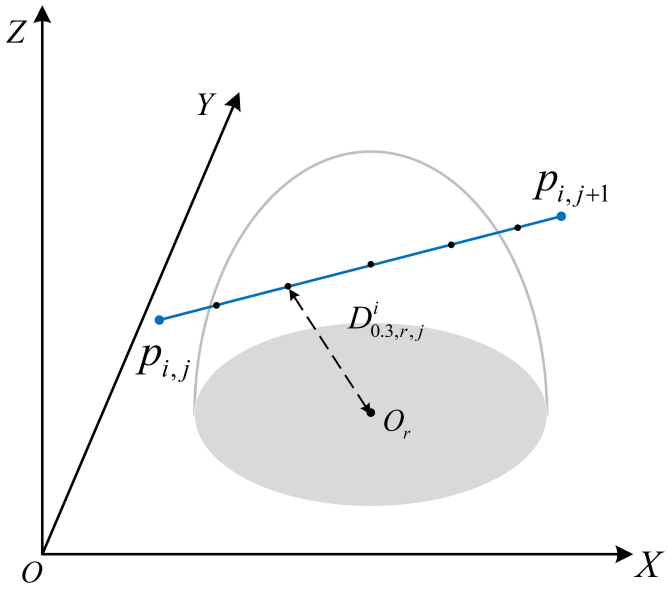


Figure 3. Computation of threat cost.

(2) No-fly zones

Assume that there are K_2 no-fly zones in the environment. The u-th no-fly zone is denoted by J_u, where $u \in \{1, 2, \cdots, K_2\}$. Each no-fly zone J_u is modeled as a vertical cylindrical forbidden region characterized by its horizontal center (x_u^{NFZ}, y_u^{NFZ}), radius R_u^{NFZ}, and height interval $[z_{\min}^u, z_{\max}^u]$. The cylindrical region is defined as:

$$J_u = \left\{ (x, y, z) \in \mathbb{R}^3 \middle| (x - x_u^{NFZ})^2 + (y - y_u^{NFZ})^2 \leqslant (R_u^{NFZ})^2, z_{\min}^u \leqslant z \leqslant z_{\max}^u \right\}. \tag{16}$$

For a given flight segment $L_{i,j}$, we say that it enters the NFZ J_u if and only if $L_{i,j} \cap J_u \neq \emptyset$. The corresponding NFZ penalty for segment $L_{i,j}$ is computed as [36]

$$\omega_{j,u}^i := \begin{cases} P_{NFZ} \times l_{i,j}, & L_{i,j} \cap J_u \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

$$f_N(\vec{L}_{i,j}) := \sum_{u=1}^{K_2} \omega_{j,u}^i, \tag{18}$$

$$F_{NFZ}(Path) := \sum_{i=1}^{N} \sum_{j=0}^{M} f_N(\vec{L}_{i,j}), \tag{19}$$

where P_{NFZ} denotes the penalty coefficient for no-fly zones, and $l_{i,j}$ specifies the length of flight segment $\vec{L}_{i,j}$ that passes through the no-fly zone J_k.

The comprehensive threat cost is formulated as:

$$F_4(Path) := F_{threat}(Path) + F_{NFZ}(Path) \tag{20}$$

3.5. Terrain Collision Threat Cost

In order to avoid collisions with mountains, when navigating in mountainous terrain, the UAV's flight altitude should always remain above the underlying terrain. The terrain threat avoidance constraint is modeled using a 2D Gaussian-like function, whose mathematical form is given as follows [15]:

$$D_{i,j} := z_{i,j} - Z(x_{i,j}, y_{i,j}) \tag{21}$$

$$T_{i,j} := \begin{cases} \infty, & D_{i,j} < 0 \\ \exp\left[-\frac{1}{2} \cdot \left(\frac{D_{i,j}^2}{\rho^2}\right)^C\right], & 0 < D_{i,j} \leqslant h_{safe} \\ 0, & D_{i,j} > h_{safe} \end{cases} \tag{22}$$

where $Z(x_{i,j}, y_{i,j})$ denotes the terrain elevation at the coordinates $(x_{i,j}, y_{i,j})$; $D_{i,j}$ denotes the altitude difference between the path point $p_{i,j}$ and the terrain at that point. h_{safe} denotes the safe-altitude threshold. In Equation (22), ρ is a distance scaling parameter used to regulate the rate at which the threat cost decreases. The parameter C is a positive integer shape factor that controls the curvature of the cost function's decay in this interval. In general, the values of parameters of ρ and C are typically fixed at 3 and 1, respectively. Finally, $T_{i,j}$ denotes the terrain threat cost at the path point $p_{i,j}$. Thus, the terrain collision threat cost is derived using the following:

$$F_5(Path) := \sum_{i=1}^{N} \sum_{j=1}^{M} T_{i,j}, \tag{23}$$

3.6. Multi-UAV Separation Maintenance Cost

In multi-UAV path planning, spatial collisions between any two UAVs must be prevented. To ensure collision avoidance, a cost is imposed on maintaining sufficient separation among UAVs, effectively serving as a guarantee of adequate distance. This requires comparing every waypoint of each UAV in the formation with every waypoint of the other UAVs. It is assumed that all UAVs depart simultaneously and maintain constant speed during flight.

For any two UAVs u and v, with $u, v \in [1, N]$ and $u \neq v$, a safe distance D_{safe} is prescribed between them. By traversing all waypoints along the flight paths of UAV u and v, a penalty is assessed for the safety-distance cost whenever the spatial separation between the a-th waypoint of UAV u and the b-th waypoint of UAV v drops below D_{safe}. The multi-UAV separation maintenance cost is calculated as follows:

$$Cost_{u,a,v,b} = \begin{cases} P_{col}, & \text{if } \|p_{u,a}p_{v,b}\| < D_{safe} \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

$$F_6(Path) = \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \sum_{a=1}^{M} \sum_{b=1}^{M} Cost_{u,a,v,b} \tag{25}$$

where P_{col} represents the penalty term, and $\|p_{u,a}p_{v,b}\|$ denotes the distance between waypoints $p_{u,a}$ and $p_{v,b}$.

3.7. Overall Cost Function

For multi-UAV trajectory planning, the primary goal is to determine efficient routes for all UAVs. Consistent with the preceding discussion, the overall cost function is defined as:

$$F_{total}(Path) = \sum_{k=1}^{6} w_k F_k(Path), \tag{26}$$

where w_k represents the weight assigned to the k-th cost component. These weights can be adapted to account for varying terrain features or specific task requirements.

4. The Proposed RLHSSA Algorithm

Motivated by the persistent challenges in multi-UAV cooperative path planning, this study introduces a novel hybrid optimization algorithm, namely the Reinforcement Learning-guided Hybrid Sparrow Search Algorithm (RLHSSA). The proposed algorithm integrates an improved Sparrow Search Algorithm (SSA), an adaptive differential perturbation strategy, and an adaptive Levy flight strategy. These three distinct operators are dynamically scheduled by a reinforcement learning framework, enabling the algorithm to adapt its search behavior throughout the optimization process.

4.1. Standard SSA

The sparrow population is divided into two main groups: producers, who are responsible for finding food-rich areas, and scroungers, who follow the producers. Additionally, a subset of the population acts as sentinels to alert the swarm of any danger [37,38].

In the SSA, each sparrow's position represents a candidate solution in the search space. The algorithm's core mechanics are defined by the position update rules for these different roles.

The position of producers is updated based on the perceived environmental safety. This behavior is described as follows

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \times \exp\left(\frac{-i}{\alpha \times t_{\max}}\right) & \text{if } R_2 < ST \\ X_{i,j}^{t} + Q \times L & \text{if } R_2 \geq ST \end{cases} \tag{27}$$

where t is the current iteration, $t_{\max}$ is the maximum number of iterations, and $i = 1, \cdots, n$, $j = 1, \cdots, d$. $X_{i,j}$ is the position of the i-th sparrow in the j-th dimension, $\alpha \in (0, 1]$ is a random number, Q is a random number following a normal distribution, and L is a $1 \times d$ matrix of ones. $R_2 \in [0, 1]$ and $ST \in [0.5, 1.0]$ represent the alarm value and the safety threshold, respectively.

Scroungers update their positions by either following the best producer or foraging elsewhere if they are "hungry" (i.e., have poor fitness). The update rule is:

$$X_{i,j}^{t+1} = \begin{cases} Q \times \exp\left(\frac{X_{worst}^{t} - X_{i,j}^{t}}{i^2}\right) & \text{if } i > n/2 \\ X_{p}^{t+1} + |X_{i,j}^{t} - X_{p}^{t+1}| \times A^{+} \times L & \text{otherwise} \end{cases} \tag{28}$$

where X_p is the optimal position found by the producers, X_{worst} is the current global worst position, and A is a matrix with elements randomly assigned to 1 or -1.

Sparrows aware of danger (sentinels) update their positions to move towards the best current solution or closer to their neighbors, which is modeled as:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^{t} + \beta \times |X_{i,j}^{t} - X_{best}^{t}| & \text{if } f_i > f_g \\ X_{i,j}^{t} + K \times \left(\frac{|X_{i,j}^{t} - X_{worst}^{t}|}{(f_i - f_w) + \epsilon}\right) & \text{if } f_i = f_g \end{cases} \tag{29}$$

where X_{best} is the current global best position, β is a step size control parameter, $K \in [-1, 1]$ is a random number, and f_i, f_g, and f_w are the fitness values of the current sparrow, the global best, and the global worst, respectively.

4.2. Proposed Hybrid Optimization Strategies

To overcome the limitations of the standard SSA, we introduce three distinct and powerful optimization operators designed to balance exploration and exploitation.

4.2.1. Elite-SSA

This operator is an enhanced version of the standard SSA, designed to accelerate convergence and refine solutions with high precision. Its update rules are stratified based on the sparrow's role. The top-performing individuals (elite producers) are guided by the global best solution to enhance exploitation:

$$X_i(t+1) = X_{best}(t) + \mu \times \mathcal{N}(0,1) \times (pBest_i(t) - X_i(t)), \tag{30}$$

where $pBest_i(t)$ is the personal best position of sparrow i, $X_{best}(t)$ is the global best position, $\mathcal{N}(0,1)$ represents a standard normal distribution, and $\mu \in [0,1]$ is a scaling factor.

Other producers follow a modified update rule dependent on the safety value R_2:

$$X_i(t+1) = \begin{cases} pBest_i(t) \times \exp(-i/(\rho \times t_{\max})) & \text{if } R_2 < ST \\ pBest_i(t) + \mathcal{N}(0,1) & \text{if } R_2 \geq ST \end{cases} \tag{31}$$

where ρ is a random decay scale factor.

Followers also have a dual-strategy update:

$$X_i(t+1) = \begin{cases} \mathcal{N}(0,1) \times \exp\left(\frac{X_{worst}(t) - pBest_i(t)}{i^2}\right) & \text{if } i > n/2 \\ X_{best}(t) + |pBest_i(t) - X_{best}(t)| \times A^+ & \text{otherwise} \end{cases} \tag{32}$$

Additionally, the behavior of sentinels is retained from the standard SSA to help the population evade local optima. This anti-predation mechanism is modeled as:

$$X_i(t+1) = \begin{cases} X_{best}(t) + \beta \times |X_i(t) - X_{best}(t)| & \text{if } f_i > f_g \\ X_i(t) + K \times \left(\frac{|X_i(t) - X_{worst}(t)|}{(f_i - f_w) + \epsilon}\right) & \text{if } f_i = f_g \end{cases} \tag{33}$$

4.2.2. Adaptive Differential Perturbation Strategy

The core idea is to introduce diversity into the population by leveraging difference vectors among randomly selected individuals, with guidance from the current optimal solution. The formulation of this strategy is given below:

$$\begin{aligned} v_i(t+1) = & X_{g_1}(t) + rand \times (P_{best}(t) - X_{g_1}(t)) \\ & + F_r \times (X_{g_2}(t) - X_{g_3}(t)). \end{aligned} \tag{34}$$

Here, g_1, g_2, and g_3 represent three unique integers from the current population, and $P_{best}(t)$ denotes the position vector of the best-performing individual at iteration t. F_r is the scaling factor, typically within the range [0, 2]. This formulation creates a mutant vector $v_i(t+1)$ that is guided by the best solution while still incorporating differential information from the broader population to maintain diversity.

Following the mutation step, a crossover operation is performed to generate the trial vector, which increases the potential diversity of the perturbed parameters.

4.2.3. Adaptive Levy Flight Strategy

The Levy flight strategy constitutes a random trajectory defined by a sequence of short movements interspersed with occasional long jumps. This behavior has been observed in the foraging patterns of various animals and insects, and it has been shown to be an effective strategy for exploring complex and unknown search spaces. In the context of optimization algorithms, incorporating Levy flight can enhance the algorithm's capacity to bypass local optima and identify promising regions far from existing solutions.

The corresponding step size s is computed as follows:

$$s = \frac{U}{|V|^{1/\lambda}}, \tag{35}$$

where U and V are random numbers sampled from normal distributions $N(0, \sigma_u^2)$ and $N(0, \sigma_v^2)$, respectively. λ is the stability index and its value typically ranges from 0 to 2. In this study, we set λ to 1.5, a common value used in the literature for simulating Levy flights in optimization algorithms [39,40].

The parameters σ_u and σ_v are related to the variances of the normal distributions and can be calculated as follows: σ_v is typically set to 1, and σ_u is determined by:

$$\sigma_u = \left(\frac{\Gamma(1+\lambda) \times \sin(\frac{\pi\lambda}{2})}{\Gamma(\frac{1+\lambda}{2}) \times \lambda \times 2^{\frac{\lambda-1}{2}}} \right)^{1/\lambda}, \tag{36}$$

where $\Gamma(\cdot)$ is the Gamma function. This formula ensures that the generated step sizes follow a Levy distribution with the chosen stability index λ.

To introduce adaptivity into the Levy flight strategy, we incorporate a non-linear parameter p that changes with the iteration number t. This parameter influences the magnitude of the Levy flights during the optimization process. The non-linear parameter p is defined as:

$$p = 2 \times \cos\left(\frac{\pi}{3} \times \left(\frac{t}{t_{\max}} \right)^2 \right) - 1. \tag{37}$$

This non-linear function allows the Levy flight strategy to adapt its influence throughout the optimization process. The value of p transitions from 1 towards -1, promoting greater exploration with larger steps in initial iterations and smaller, refining steps in subsequent ones.

The Levy flight strategy is used to generate a new candidate position $LevyX_i(t+1)$ for the i-th individual based on its current position $X_i(t)$ and the position of the best individual found so far, $P_{best}(t)$. The update rule is given by:

$$LevyX_i(t+1) = X_i(t) + p \times s \times (P_{best}(t) - X_i(t)). \tag{38}$$

4.2.4. Reinforcement Learning-Based Strategy Scheduling

Within the domain of cooperative pathfinding for multiple UAVs, the balancing of exploration and exploitation plays a vital role. Relying solely on a single search strategy, whether focused on broad exploration or fine-grained exploitation, may not be sufficient to navigate the intricate search space and yield optimal solutions efficiently.

Reinforcement learning offers a promising framework for enabling algorithms to make intelligent decisions from real-time environmental feedback. By learning through trial and error, an RL agent can adapt its behavior to maximize a cumulative reward. This capability is particularly well-suited for dynamically adjusting the search strategy in optimization algorithms, ensuring a more robust and adaptive search process that optimizes multiple objectives, including path length, obstacle avoidance, and cooperation among UAVs.

Q-learning, a model-free reinforcement learning algorithm, is employed in this work to learn an optimal policy for dynamically selecting among the three proposed operators: the Elite Sparrow Search Algorithm (Section 4.2.1), the Adaptive Differential Perturbation Strategy (Section 4.2.2), and the Adaptive Levy Flight Strategy (Section 4.2.3). The goal is to leverage the unique strengths of each operator at different stages of the optimization process.

To implement this, we establish a 10×3 Q-table, where each Q-value represents the expected future reward for taking a specific action in a given state. The optimization process is divided into 10 distinct states based on the current iteration count. Specifically, the total number of iterations is partitioned into 10 equal intervals. If the current iteration t falls within the k-th interval, the algorithm is considered to be in state s_k, where $k \in \{1, 2, \cdots, 10\}$. Each state corresponds to a different phase of the search, allowing the algorithm to learn which strategy is most effective at different stages. The agent in our Q-learning framework has three possible actions at each state:

- Applying the Elite Sparrow Search Algorithm (a_1).
- Applying the Adaptive Differential Perturbation Strategy (a_2).
- Applying the Adaptive Levy Flight Strategy (a_3).

The Q-table's structure is illustrated in Table 1.

Table 1. The Q-table framework for strategy scheduling.

State	**Action**		
	a_1**: Elite-SSA**	a_2**: Diff. Perturbation**	a_3**: Levy Flight**
s_1	$Q(s_1, a_1)$	$Q(s_1, a_2)$	$Q(s_1, a_3)$
s_2	$Q(s_2, a_1)$	$Q(s_2, a_2)$	$Q(s_2, a_3)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
s_{10}	$Q(s_{10}, a_1)$	$Q(s_{10}, a_2)$	$Q(s_{10}, a_3)$

The Q-values in the Q-table are updated iteratively using the standard Q-learning update rule:

$$Q(s_t, a_t) \leftarrow (1 - \alpha_{\mathrm{RL}}) Q(s_t, a_t) + \alpha_{\mathrm{RL}} \left[r_{t+1} + \gamma_{\mathrm{RL}} \max_a Q(s_{t+1}, a) \right] \tag{39}$$

where $\alpha_{\mathrm{RL}} \in [0, 1]$ is the learning rate, r_{t+1} is the reward received after taking action a_t in state s_t, and $\gamma_{\mathrm{RL}} \in [0, 1]$ is the discount factor that balances immediate and future rewards. Initially, all Q-values are set to 0.

For achieving a trade-off between exploration and exploitation in the strategy selection process, an ϵ-greedy approach is utilized. At state s_t, an action is chosen randomly with a probability of ϵ; conversely, the action yielding the highest current Q-value is selected with a probability of $1 - \epsilon$:

$$action_{next} = \begin{cases} action_{random} & \text{if rand} < \epsilon \\ \underset{a \in \{a_1, a_2, a_3\}}{\text{argmax}} \left[Q(s_t, a) \right] & \text{otherwise} \end{cases} \tag{40}$$

The value of ϵ is typically set high initially and annealed over time to favor exploitation in later stages.

The reward r at iteration t is defined based on the relative improvement in the global best fitness value F_g:

$$r_{t+1} = \frac{F_g(t) - F_g(t+1)}{F_g(t) + \varepsilon}, \tag{41}$$

where ε denotes a small constant. This reward mechanism guides the Q-learning agent to discern which strategy is more likely to yield better solutions, thereby optimizing the overall search trajectory across various optimization phases.

4.3. Integration of the RLHSSA Algorithm

The developed RLHSSA approach is a sophisticated hybrid algorithm that integrates three distinct optimization operators: the Elite Sparrow Search Algorithm (Elite-SSA), the Adaptive Differential Perturbation strategy, and the Adaptive Levy Flight strategy. Unlike sequential enhancement models, the RLHSSA employs a dynamic scheduling framework powered by reinforcement learning to govern the entire search process.

In each iteration, a Q-learning agent first determines the current phase of the optimization (the state) and then selects one of the three operators to be executed [41]. This choice is based on a learned policy that aims to maximize long-term rewards, effectively balancing the global exploration and local exploitation needs of the search. After the chosen operator updates the population's positions, fitness is evaluated, and a reward based on the performance improvement is fed back to the Q-learning agent to update its policy.

The pseudocode of the RLHSSA is given below in Algorithm 1.

Algorithm 1 Pseudocode of the RLHSSA Algorithm

Initialize
$t \leftarrow 1$
Randomly generate initial population positions $X_i,\ i = 1, \cdots, NP$
Set parameters: $NP, ND, t_{\max}, F_r, \lambda, \epsilon, \alpha_{\mathrm{RL}}, \gamma_{\mathrm{RL}}$
Define action set $\mathcal{A} = \{a_1, a_2, a_3\}$
Initialize Q-table $Q(s, a) = 0$ for all $s \in \{1, \ldots, 10\},\ a \in \mathcal{A}$
Compute initial fitness $f(X_i)$ for each individual i
Initialize personal bests $pBest_i \leftarrow X_i$ and find the global best P_{best}
Optimize
while $t \leq t_{\max}$ **do**
    Determine current state $s_t \leftarrow \lceil 10 \times (t/t_{\max}) \rceil$
    Choose action $a_t \in \mathcal{A}$ via ϵ-greedy policy on $Q(s_t, \cdot)$ using Equation (40)
    // **Dynamically Scheduled Operator Execution**
    if $a_t = a_1$ **then**
        Perform Elite Sparrow Search Algorithm (Elite-SSA) using Equations (30)–(33)
    else if $a_t = a_2$ **then**
        Perform Adaptive Differential Perturbation using Equation (34)
    else
        Perform Adaptive Levy Flight strategy using Equations (35)–(38)
    end if
    Evaluate fitness of the new population
    Update personal bests ($pBest_i$) and the global best (P_{best})
    // **Reinforcement Learning Update**
    Compute reward r_{t+1} using Equation (41)
    Observe next state s_{t+1}
    Adjust Q-value $Q(s_t, a_t)$ using Equation (39)
    $t \leftarrow t + 1$
end while
Return best solution P_{best}

4.4. Cooperative Multi-UAV Path Planning with RLHSSA

Applying the proposed RLHSSA algorithm for addressing multi-UAV cooperative path planning within mountainous environments involves several key steps. The process is outlined below:

Step 1: Establish the multi-UAV path planning scenario. This step includes specifying UAV parameters, initial and target positions, environmental information (terrain, obstacles), and flight constraints.

Step 2: Initialize multi-UAV path representation and population. A set of paths for all UAVs is represented as a solution vector to define the algorithm's search space. The

initial population of candidate solution vectors (path sets) is randomly generated within this space, and their initial quality is evaluated using a problem-specific fitness function.

Step 3: Optimize the path planning model using RLHSSA. The RLHSSA algorithm iteratively refines the population of path sets over a set number of iterations. During each iteration, the algorithm updates the path sets guided by fitness evaluation, leveraging its dynamically scheduled exploration and exploitation mechanisms to improve solution quality.

Step 4: Provide the best-performing solution. Upon reaching the termination criterion (e.g., the maximum number of iterations), the algorithm concludes the optimization and outputs the best solution found throughout the search, representing the final set of optimized cooperative paths for the multi-UAV system.

5. Simulation Example

This section details the experimental evaluation of the proposed RLHSSA's performance in multi-UAV cooperative path planning in mountainous environments. A range of complex simulation scenarios were established to assess the algorithm's effectiveness. This mountainous environment is widely adopted in UAV path-planning studies, as its steep elevation variations, irregular obstacles, and complex threat distributions provide a highly challenging three-dimensional testbed for evaluating cooperative maneuverability and obstacle/threat avoidance.

The simulations were implemented in MATLAB R2022a and executed on a workstation equipped with an Intel Core i7-9700 CPU (3.00 GHz) and 16 GB of RAM, running the Windows 10 operating system.

5.1. Simulation Environment and Experimental Setup

To thoroughly evaluate the proposed algorithm's capabilities under varying complexities and safety requirements, two distinct simulation scenarios were designed: a threat-free environment and a threatened environment. The parameters defining the missions for the three UAVs in both simulation scenarios are shown in Table 2.

Table 2. Parameters of multiple UAVs.

	UAV No.	Start Point	Target Point
Scenario 1	1	(0, 100, 5)	(200, 60, 10)
	2	(0, 120, 8)	(200, 70, 12)
	3	(0, 140, 10)	(200, 80, 12)
Scenario 2	1	(0, 40, 8)	(200, 68, 8)
	2	(0, 100, 10)	(200, 75, 6)
	3	(0, 140, 12)	(200, 85, 7)

Furthermore, for the threatened environment scenario (Scenario 2), specific threats and NFZs are introduced to model a more complex and dangerous operational area. The details regarding these threat elements are provided in Table 3. For comparison, the proposed RLHSSA algorithm is compared with Multi-Strategy Improved Differential Evolution (MSIDE) algorithm [26], hybrid PSO (SDPSO) algorithm [42], Hybrid GWO and Differential Evolution (HGWODE) algorithm [43]. Table 4 lists the key parameters for all algorithms used in the simulations.

In the comparative experiments, Table 5 enumerates the key parameters of the proposed RLHSSA algorithm. For impartial comparison, parameters for all competing algorithms were aligned with those of RLHSSA. Furthermore, 30 independent trials were conducted for each algorithm on each simulation case to address the inherent randomness of the optimization process. Path quality was assessed using the comprehensive fitness

function, $F_{total}(Path)$, defined in Equation (26). Subsequently, the B-spline method was employed for flight path smoothing [44].

Table 3. Information of threat and NFZs in Scenario 2.

Threat/NFZ	Threat/NFZ Center	Radius	Height	Threat Level
Threat	(50, 30, 0)	20	–	10
Threat	(100, 100, 0)	18	–	10
NFZ	(55, 140)	20	80	–
NFZ	(165, 45)	18	80	–

Table 4. Algorithm parameter setting.

Algorithm	Parameter
MSIDE	$F_{\min} = 0$, $F_{\max} = 1$, $CR = 0.8$
SDPSO	$w : 0.4 \sim 0.9$, $c_1 = c_2 : 0.4 \sim 2.0$
HGWODE	$a = 2 \sim 0$, $C_r = 0.5$
RLHSSA	$ST = 0.8$, $\mu = 0.5$, $\lambda = 1.5$,

Table 5. Parameter definitions and values.

Parameter	Description	Value
$t_{\max}$	Maximum iteration count	2000
NP	Population count	50
M	waypoint count	7
$[H_{\min}, H_{\max}]$	UAV operational altitude range	[1, 100]
$\vartheta_{\max}$	Max turning angle	30°
$\xi_{\max}$	Max climbing angle	30°
h_{safe}	safe-altitude threshold	15
D_{safe}	Inter-UAV safe distance	20

5.2. Simulation Results in Scenario 1

This section presents the simulation results for the threat-free environment (Scenario 1). Figures 4 and 5 visually display the 3D and top views of the cooperative paths planned by the four algorithms in Scenario 1. From these figures, it is evident that while all four algorithms can generate collision-free and safe paths for multiple UAVs, significant disparities exist in the quality of the planned trajectories. Specifically, RLHSSA generates feasible and high-quality flight paths for each UAV, whereas the paths produced by MSIDE, SDPSO, and HGWODE are noticeably inferior in terms of safety and path length.

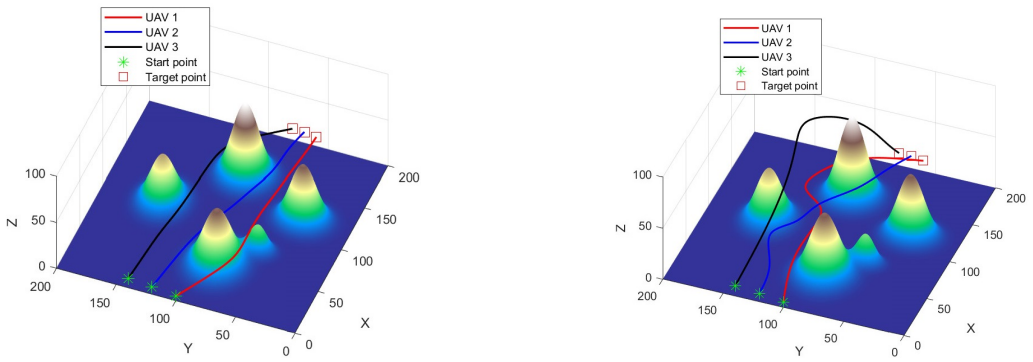


Figure 4. *Cont.*

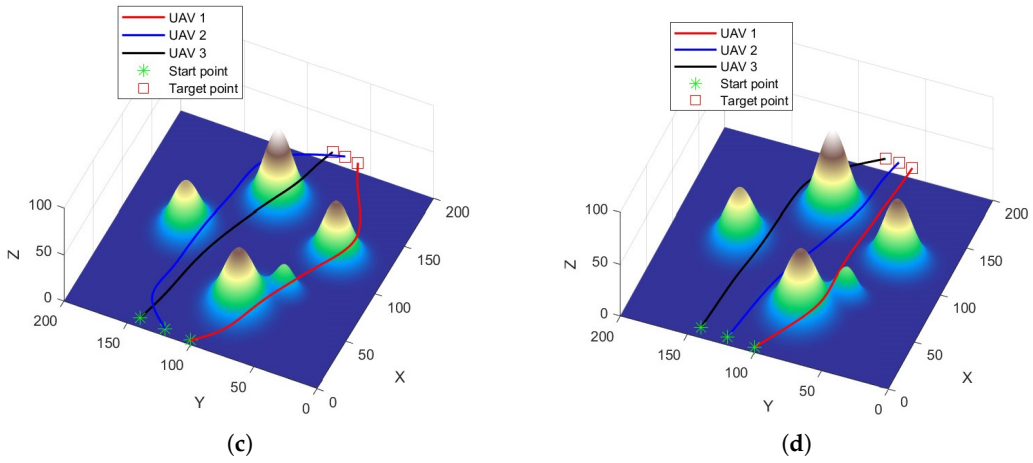


Figure 4. 3D flight paths visualization of three UAVs in Scenario 1: (**a**) MSIDE; (**b**) SDPSO; (**c**) HGWODE; (**d**) RLHSSA.

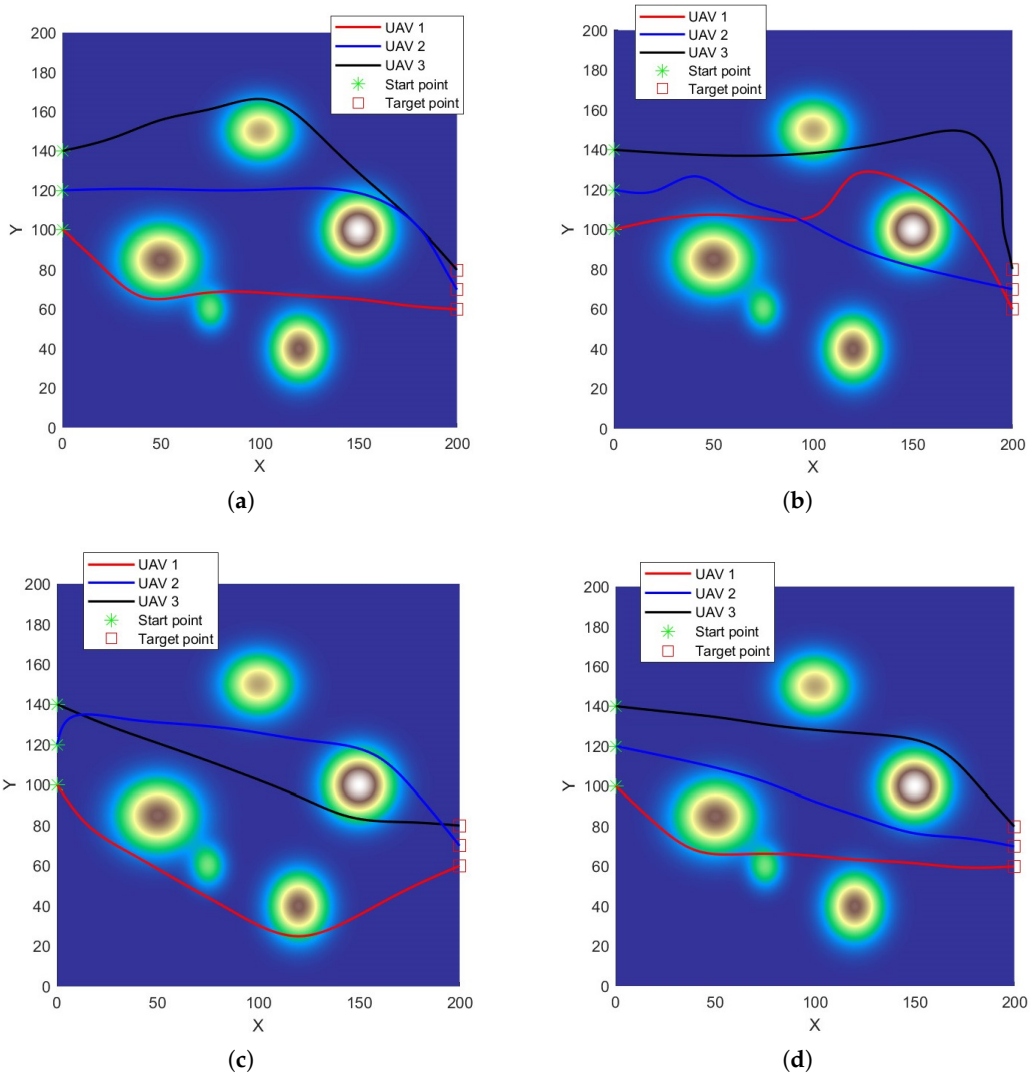


Figure 5. Three UAVs flight trajectories (top view) in Scenario 1: (**a**) MSIDE; (**b**) SDPSO; (**c**) HGWODE; (**d**) RLHSSA.

Furthermore, Figure 6 displays the convergence curves for the four algorithms in Scenario 1. This figure illustrates the correlation between the objective function value and the number of iterations, clearly showing that RLHSSA exhibits faster convergence

speed and achieves a significantly lower final objective function value compared to MSIDE, SDPSO, and HGWODE. This indicates that RLHSSA is capable of efficiently finding optimal or near-optimal solutions in this scenario.

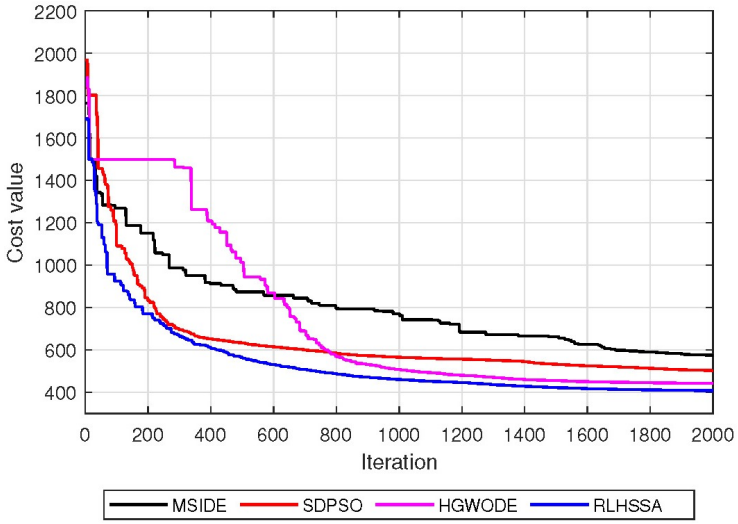


Figure 6. Convergence curves of four algorithms in Scenario 1.

Statistical metrics from 30 independent runs in Scenario 1 are summarized in Table 6. These metrics include the minimum (Best), maximum (Worst), mean fitness (Mean), the standard deviation (Std) of the fitness values, the average computational time (Mean Time), and the success rate (SR). SR denotes the percentage of successful runs that achieve feasible and constraint-satisfying paths.

As shown in Table 6, RLHSSA consistently outperforms all comparison algorithms across the metrics of solution quality and reliability. RLHSSA achieves the lowest Best, Worst, and Mean values, highlighting its exceptional ability to find optimal solutions and avoid convergence to local optima. Furthermore, the significantly lower Std value for RLHSSA indicates its superior stability and robustness across multiple runs, suggesting that its performance is less susceptible to the stochastic nature of the optimization process. The highest SR for RLHSSA further confirms its reliability in consistently finding feasible and valid paths in the threat-free environment. Crucially, the "Mean Time (s)" metric shows that the computational times for all four algorithms are comparable. This demonstrates that RLHSSA achieves its superior performance without introducing significant computational overhead.

Table 6. Experimental outcomes in Scenario 1 across 30 independent runs.

NO.	Indicators	MSIDE	SDPSO	HGWODE	RLHSSA
1	Best	430.0323	410.2035	405.4126	396.3353
2	Worst	586.0367	485.3387	472.5805	416.0802
3	Mean	468.7183	434.4157	426.9850	405.0570
4	Std	43.1114	16.7519	12.4036	4.0826
5	SR (%)	70.0	83.33	90.0	100.0
6	Mean Time (s)	12.48	11.87	13.32	12.81

5.3. Simulation Results in Scenario 2

To assess the proposed algorithm's performance under more challenging conditions, simulation experiments were conducted in the threatened environment (Scenario 2), which includes various threats and no-fly zones. Figures 7 and 8 visually present the 3D and top views of the cooperative paths planned by the four algorithms in Scenario 2. Observation

of these figures highlights significant differences in the quality and safety of the paths produced by the different algorithms. Under this challenging flight environment, RLHSSA demonstrates excellent performance, planning the safest and shortest flight paths for each UAV. In contrast, the other algorithms exhibit various limitations in generating optimal paths in this challenging scenario. MSIDE appears to have problems in path planning in this environment; its planned trajectories cannot guarantee the safety of each UAV passing through this area, indicating a possibility of collisions. While SDPSO and HGWODE are capable of providing safe and feasible paths, their generated trajectories are noticeably longer and less smooth compared to RLHSSA. This increased path length may lead to longer flight time and higher energy consumption.

Figure 9 displays the convergence trends of the four methods in Scenario 2, illustrating their search efficiency and solution-finding capabilities. As observed from the figure, all algorithms demonstrate a decreasing trend in cost value with increasing iterations, indicating their convergence towards a solution. However, significant differences in convergence speed and final solution quality are evident among the algorithms. The proposed RLHSSA algorithm exhibits the fastest convergence rate, achieving the lowest cost value within 2000 iterations, which highlights its superior search efficiency and global optimization capability in this complex scenario.

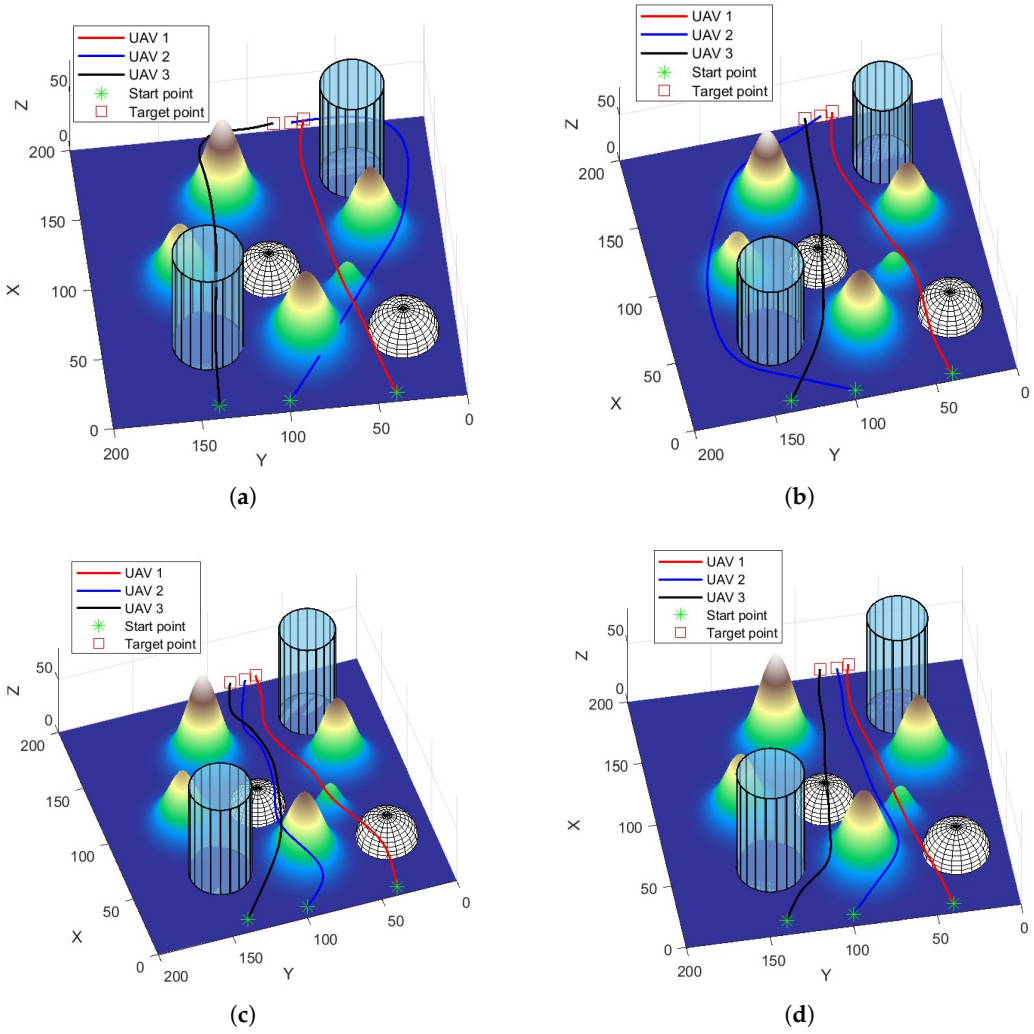


Figure 7. Three-dimensional flight path visualization of three UAVs in Scenario 2: (**a**) MSIDE; (**b**) SDPSO; (**c**) HGWODE; (**d**) RLHSSA.

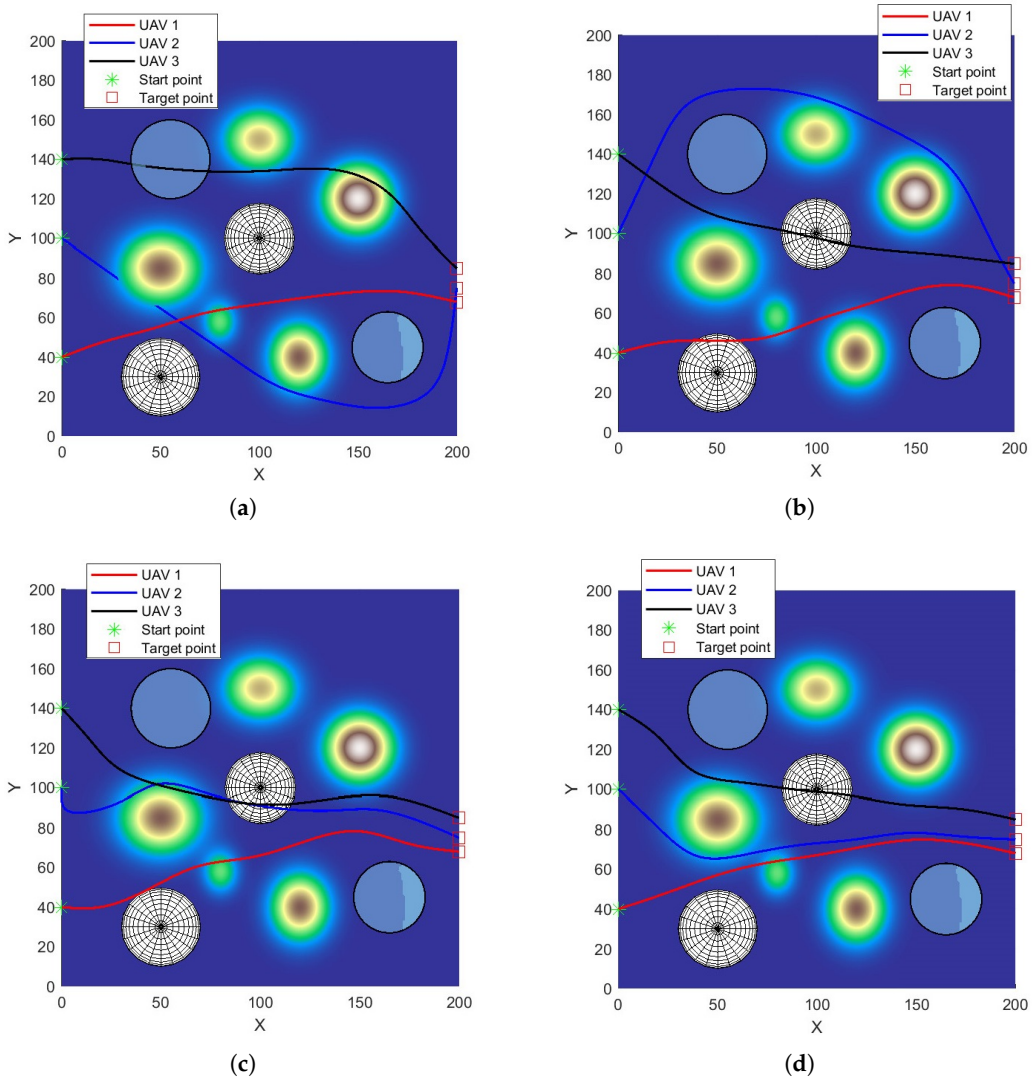


Figure 8. Three UAVs' flight trajectories (top view) in Scenario 2: (**a**) MSIDE; (**b**) SDPSO; (**c**) HGWODE; (**d**) RLHSSA.

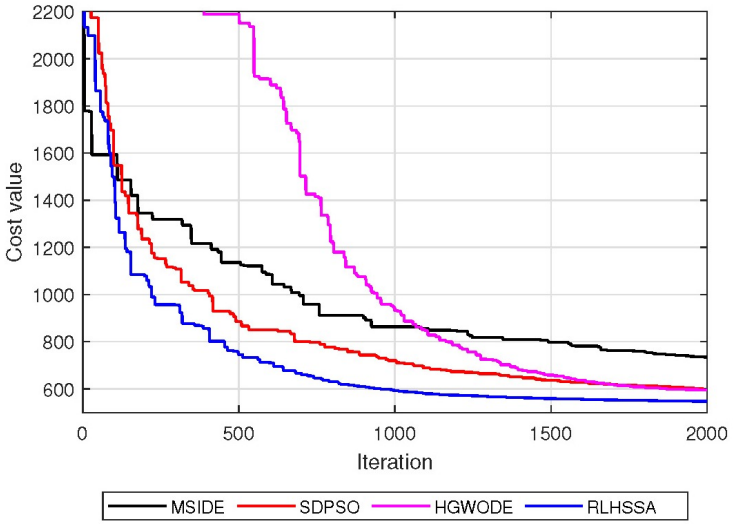


Figure 9. Convergence curves of four algorithms in Scenario 2.

Table 7 summarizes the statistical outcomes of the four algorithms after 30 runs in Scenario 2, demonstrating that the proposed RLHSSA algorithm consistently achieves the best performance across all indicators. It records the lowest mean objective value of 581.4147, demonstrating its superior ability to find high-quality solutions. Furthermore, with a low standard deviation of 8.4732 and a 100% success rate, RLHSSA exhibits remarkable stability and robustness in navigating the complex and constrained environment. Crucially, the "Mean Time (s)" metric again shows that the computational times for all algorithms are comparable. This confirms that the superior performance of RLHSSA is achieved without imposing any significant computational penalty, even in this more complex threatened environment.

Table 7. Experimental outcomes in Scenario 2 across 30 independent runs.

NO.	Indicators	MSIDE	SDPSO	HGWODE	RLHSSA
1	Best	625.3503	583.6828	585.7339	570.1830
2	Worst	920.1053	832.2043	803.0355	609.9120
3	Mean	737.2827	648.8387	625.1940	581.4147
4	Std	87.8027	62.7768	48.3840	8.4732
5	SR (%)	60.0	76.67	86.67	100.0
6	Mean Time (s)	15.38	13.83	15.74	14.28

6. Conclusions

This study addresses the intricate challenge of cooperative route planning for multiple UAVs operating in mountainous regions, with particular attention to threat presence. To solve this challenge, we propose a path planning approach that employs the RLHSSA, a novel hybrid optimization algorithm. It facilitates the generation of optimal, safe, and efficient flight paths while effectively avoiding terrain obstacles. The proposed RLHSSA integrates three distinct optimization operators: the Elite Sparrow Search Algorithm (Elite-SSA) for refined exploitation, alongside the Adaptive Differential Perturbation and Adaptive Levy Flight strategies for robust exploration. A key innovation is the use of a Reinforcement Learning (RL) agent that dynamically schedules these operators, ensuring an intelligent balance between search breadth and depth to effectively prevent premature convergence.

Subsequently, simulation experiments were conducted in two different three-dimensional mountainous environments, comparing and evaluating the RLHSSA algorithm with three advanced methods. The experimental findings consistently indicate the superiority of RLHSSA over other methods regarding solution accuracy, convergence speed, and stability. This validates the utility of RLHSSA for cooperative path planning of multi-UAVs in mountainous environments.

Future work will focus on applying the algorithm to dynamic scenarios characterized by stochastic threats. Furthermore, the transferability of the RLHSSA framework to other multi-agent systems and complex optimization problems will be further investigated.

Author Contributions: Conceptualization, L.L.; methodology, W.S. and W.J.; software, L.L. and R.H.; validation, L.L. and W.S.; formal analysis, W.J.; investigation, L.L.; resources, W.S.; data curation, W.S. and R.H.; writing—original draft preparation, L.L.; writing—review and editing, L.L. and W.S.; visualization, W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62371375), the Shaanxi Key R&D Plan Key Industry Innovation Chain Project, the Shaanxi "Hundred Teams" Project for University-Enterprise Collaboration 2025, and the Xi'an Science and Technology Plan Project (No. 2025JH-KXGC-0007).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon request.

Conflicts of Interest: Author Wei Jia was employed by the company Xi'an ASN Technology Group Co., Ltd. The remaining 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. Skorobogatov, G.; Barrado, C.; Salamí, E. Multiple UAV systems: A survey. *Unmanned Syst.* **2020**, *8*, 149–169. [CrossRef]
2. Erdelj, M.; Natalizio, E.; Chowdhury, K.R.; Akyildiz, I.F. Help from the sky: Leveraging UAVs for disaster management. *IEEE Pervas. Comput.* **2017**, *16*, 24–32. [CrossRef]
3. Wang, K.; Yang, P.; Lv, W.S.; Zhu, L.Y.; Yu, G.M. Current status and development trend of UAV remote sensing applications in the mining industry. *Chin. J. Eng.* **2020**, *42*, 1085–1095.
4. Zhou, X.Y.; Jia, W.; He, R.F.; Sun, W. High-Precision localization tracking and motion state estimation of ground-based moving target utilizing unmanned aerial vehicle high-altitude reconnaissance. *Remote Sens.* **2025**, *17*, 735. [CrossRef]
5. Hong, Y.; Kim, S.; Kwon, Y.; Choi, S.; Cha, J. Safe and Efficient Exploration Path Planning for Unmanned Aerial Vehicle in Forest Environments. *Aerospace* **2024**, *11*, 598. [CrossRef]
6. Yu, X.; Luo, W. Reinforcement learning-based multi-strategy cuckoo search algorithm for 3D UAV path planning. *Expert Syst. Appl.* **2023**, *223*, 119910. [CrossRef]
7. Qadir, Z.; Zafar, M.H.; Moosavi, S.K.R.; Le, K.N.; Mahmud, M.A.P. Autonomous UAV path-planning optimization using metaheuristic approach for predisaster assessment. *IEEE Internet Things* **2021**, *9*, 12505–12514. [CrossRef]
8. Abdel-Basset, M.; Mohamed, R.; Sallam, K.M.; Hezam, I.M.; Munasinghe, K.; Jamalipour, A. A multiobjective optimization algorithm for safety and optimality of 3-D route planning in UAV. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 3067–3080. [CrossRef]
9. Chen, J.; Zhang, R.; Zhao, H.; Li, J.; He, J. Path planning of multiple unmanned aerial vehicles covering multiple regions based on minimum consumption ratio. *Aerospace* **2023**, *10*, 93. [CrossRef]
10. Zhang, Z.; Zhu, L. A Review on Unmanned Aerial Vehicle Remote Sensing: Platforms, Sensors, Data Processing Methods, and Applications. *Drones* **2023**, *7*, 398. [CrossRef]
11. He, W.J.; Qi, X.G.; Liu, L.F. A novel hybrid particle swarm optimization for multi-UAV cooperate path planning. *Appl. Intell.* **2021**, *51*, 7350–7364. [CrossRef]
12. Xu, L.; Cao, X.B.; Du, W.B.; Li, Y.M. Cooperative path planning optimization for multiple UAVs with communication constraints. *Knowl.-Based Syst.* **2023**, *260*, 110164. [CrossRef]
13. Pan, Z.; Zhang, C.; Xia, Y.; Xiong, H.; Shao, X. An improved artificial potential field method for path planning and formation control of the multi-UAV systems. *IEEE T Circuits-II* **2021**, *69*, 1129–1133. [CrossRef]
14. Wang, W.T.; Li, X.L.; Tian, J. UAV formation path planning for mountainous forest terrain utilizing an artificial rabbit optimizer incorporating reinforcement learning and thermal conduction search strategies. *Adv. Eng. Inform.* **2024**, *62*, 102947. [CrossRef]
15. Sun, B.; Niu, N. Multi-AUVs cooperative path planning in 3D underwater terrain and vortex environments based on improved multi-objective particle swarm optimization algorithm. *Ocean Eng.* **2024**, *311*, 118944. [CrossRef]
16. Xu, C.; Xu, M.; Yin, C. Optimized multi-UAV cooperative path planning under the complex confrontation environment. *Comput. Commun.* **2020**, *162*, 196–203. [CrossRef]
17. Li, Y.; Dong, X.; Ding, Q.; Xiong, Y.; Wang, T. Improved A-STAR algorithm for power line inspection UAV path planning. *Energies* **2024**, *17*, 5364. [CrossRef]
18. Wang, J.; Li, Y.; Li, R.; Chen, H.; Chu, K. Trajectory planning for UAV navigation in dynamic environments with matrix alignment Dijkstra. *Soft Comput.* **2022**, *26*, 12599–12610. [CrossRef]
19. Guo, Y.; Liu, X.; Jiang, W.; Zhang, W. HDP-TSRRT*: A time-space cooperative path planning algorithm for multiple UAVs. *Drones* **2023**, *7*, 170. [CrossRef]
20. Zhang, Z.; Li, J.; Wang, J. Sequential convex programming for nonlinear optimal control problems in UAV path planning. *Aerosp. Sci. Technol.* **2018**, *76*, 280–290. [CrossRef]
21. Wu, Y.; Liang, T.; Gou, J.; Tao, C.; Wang, H. Heterogeneous mission planning for multiple UAV formations via metaheuristic algorithms. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 3924–3940. [CrossRef]
22. Gupta, H.; Verma, O.P. A novel hybrid coyote-particle swarm optimization algorithm for three-dimensional constrained trajectory planning of unmanned aerial vehicle. *Appl. Soft Comput.* **2023**, *147*, 110776. [CrossRef]
23. Ab Wahab, M.N.; Nazir, A.; Khalil, A.; Ho, W.J.; Akbar, M.F.; Noor, M.H.M.; Mohamed, A.S.A. Improved genetic algorithm for mobile robot path planning in static environments. *Expert Syst. Appl.* **2024**, *249*, 123762. [CrossRef]

24. Li, X.; Yu, S. Three-dimensional path planning for AUVs in ocean currents environment based on an improved compression factor particle swarm optimization algorithm. *Ocean Eng.* **2023**, *280*, 114610. [CrossRef]
25. Wang, Z.; Dai, D.; Zeng, Z.; He, D.; Chan, S. Multi-strategy enhanced Grey Wolf Optimizer for global optimization and real world problems. *Cluster Comput.* **2024**, *27*, 10671–10715. [CrossRef]
26. Zhang, M.H.; Han, Y.H.; Chen, S.Y.; Liu, M.X.; He, Z.L.; Pan, N. A multi-strategy improved differential evolution algorithm for UAV 3D trajectory planning in complex mountainous environments. *Eng. Appl. Artif. Intel.* **2023**, *125*, 106672. [CrossRef]
27. Meng, K.; Chen, C.; Wu, T.; Xin, B.; Liang, M.; Deng, F. Evolutionary state estimation-based multi-strategy jellyfish search algorithm for multi-UAV cooperative path planning. *IEEE Trans. Intell. Veh.* **2024**, *10*, 2490–2507 . [CrossRef]
28. Wang, F.; Wang, X.; Sun, S. A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. *Inform. Sci.* **2022**, *602*, 298–312. [CrossRef]
29. Zhang, S.; Li, Y.; Dong, Q. Autonomous navigation of UAV in multi-obstacle environments based on a deep reinforcement learning approach. *Appl. Soft Comput.* **2022**, *115*, 108194. [CrossRef]
30. Skarka, W.; Ashfaq, R. Hybrid machine learning and reinforcement learning framework for adaptive UAV obstacle avoidance. *Aerospace* **2024**, *11*, 870. [CrossRef]
31. Niu, Y.; Yan, X.; Wang, Y.; Niu, Y. Three-dimensional collaborative path planning for multiple UCAVs based on improved artificial ecosystem optimizer and reinforcement learning. *Knowl.-Based Syst.* **2023**, *276*, 110782. [CrossRef]
32. Qu, C.; Gai, W.; Zhong, M.; Zhang, J. A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning. *Appl. Soft Comput.* **2020**, *89*, 106099. [CrossRef]
33. Jiaqi, S.; Li, T.; Hongtao, Z.; Xiaofeng, L.; Tianying, X. Adaptive multi-UAV path planning method based on improved gray wolf algorithm. *Comput. Electr. Eng.* **2022**, *104*, 108377. [CrossRef]
34. Xue, J.K.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
35. Niu, Y.B.; Yan, X.; Wang, Y.; Niu, Y.Z. Three-dimensional UCAV path planning using a novel modified artificial ecosystem optimizer. *Expert Syst. Appl.* **2023**, *217*, 119499. [CrossRef]
36. Lv, L.; Liu, H.J.; He, R.; Jia, W.; Sun, W. A Novel HGW Optimizer with Enhanced Differential Perturbation for Efficient 3D UAV Path Planning. *Drones* **2025**, *9*, 212. [CrossRef]
37. Gharehchopogh, F.S.; Namazi, M.; Ebrahimi, L.; Abdollahzadeh, B. Advances in sparrow search algorithm: A comprehensive survey. *Arch. Comput. Method. E.* **2023**, *30*, 427–455. [CrossRef]
38. He, Y.; Wang, M. An improved chaos sparrow search algorithm for UAV path planning. *Sci. Rep.* **2024**, *14*, 366. [CrossRef]
39. Amirsadri, S.; Mousavirad, S.J.; Ebrahimpour-Komleh, H. A Levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training. *Neural Comput. Appl.* **2018**, *30*, 3707–3720. [CrossRef]
40. Gong, G.M.; Fu, S.W.; Huang, H.S.; Huang, H.F.; Luo, X. Multi-strategy improved snake optimizer based on adaptive lévy flight and dual-lens fusion. *Cluster Comput.* **2025**, *28*, 268. [CrossRef]
41. Liu, H.; Zhang, X.; Zhang, H.; Li, C.; Chen, Z. A reinforcement learning-based hybrid Aquila Optimizer and improved Arithmetic Optimization Algorithm for global optimization. *Expert Syst. Appl.* **2023**, *224*, 119898. [CrossRef]
42. Yu, Z.H.; Si, Z.J.; Li, X.B.; Wang, D.; Song, H.B. A novel hybrid particle swarm optimization algorithm for path planning of UAVs. *IEEE Internet Things* **2022**, *9*, 22547–22558. [CrossRef]
43. Yu, X.B.; Jiang, N.J.; Wang, X.M.; Li, M.Y. A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning. *Expert Syst. Appl.* **2023**, *215*, 119327. [CrossRef]
44. Qu, C.; Gai, W.; Zhang, J.; Zhong, M. A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning. *Knowl.-Based Syst.* **2020**, *194*, 105530. [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.