drones

MDPI

Article

Collaborative Vehicle-Mounted Multi-UAV Routing and Scheduling Optimization for Remote Sensing Observations

Bing Du [1,2,3], Anqi Tang [1,2,3], Huping Ye [1,2,3], Huanyin Yue [1,2,3], Chenchen Xu [1,2,3], Lina Hao [1,2,3], Hongbo He [1,2,3] and Xiaohan Liao [1,2,3,*]

[1] State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China; dubing23@mails.ucas.ac.cn (B.D.); hehb.20b@igsnrr.ac.cn (H.H.)
[2] Key Laboratory of Low Altitude Geographic Information and Air Route, Civil Aviation Administration of China, Beijing 100101, China
[3] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: liaoxh@igsnrr.ac.cn

Highlights

What are the main findings?

- A multi-UAV routing and scheduling framework for remote sensing observation tasks is proposed, and comparative experiments with classical algorithms demonstrate the superiority of the proposed method.
- A Multi-Region Edge Recombination Crossover operator and an Adaptive Hybrid Mutation mechanism are designed to enhance the genetic algorithm's performance in multi-UAV task scheduling and routing optimization.

What are the implications of the main findings?

- The routing and scheduling schemes of vehicle-mounted multi-UAV systems are optimized, significantly reducing the coverage cost of remote sensing observations.
- The proposed framework provides practical scheduling guidance for large-scale remote sensing applications using vehicle-mounted multi-UAV systems, effectively minimizing task redundancy.

Abstract

Vehicle-mounted multi-UAV (VM-UAV) systems offer enhanced flexibility and rapid deployment for large-scale remote sensing tasks such as disaster response and land surveys. However, maximizing their operational efficiency remains challenging, as it requires the simultaneous resolution of task scheduling and coverage path planning—an NP-hard problem. This study presents a novel multi-objective genetic algorithm (GA) framework that jointly optimizes routing and scheduling for cost-constrained, load-balanced multi-UAV remote sensing missions. To improve convergence speed and solution quality, we introduce two innovative operators: a Multi-Region Edge Recombination Crossover (MRECX) to preserve superior path segments from parents and an Adaptive Hybrid Mutation (AHM) mechanism that dynamically adjusts mutation strategies to balance exploration and exploitation. The algorithm minimizes total flight distance while equalizing workload distribution among UAVs. Extensive simulations and experiments demonstrate that the proposed GA significantly outperforms conventional GA, particle swarm optimization (PSO), ant colony optimization (ACO), and clustering-based planning methods in both solution quality and robustness. The practical applicability of our framework is further validated through two real-world case studies. The results confirm that the proposed approach delivers an effective and scalable solution for vehicle-mounted multi-UAV scheduling and path planning, enhancing operational efficiency in time-critical remote sensing applications.

check for updates

Academic Editors: Heng Shi, Jihong Zhu, Zheng Chen and Minchi Kuang

Received: 5 October 2025
Revised: 29 October 2025
Accepted: 3 November 2025
Published: 11 November 2025

Citation: Du, B.; Tang, A.; Ye, H.; Yue, H.; Xu, C.; Hao, L.; He, H.; Liao, X. Collaborative Vehicle-Mounted Multi-UAV Routing and Scheduling Optimization for Remote Sensing Observations. *Drones* **2025**, *9*, 783. https://doi.org/10.3390/drones9110783

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/).

Keywords: unmanned aerial vehicle; scheduling optimization; genetic algorithm; coverage path planning; remote sensing; emergency management

1. Introduction

Unmanned Aerial Vehicles (UAVs) have become a critical component of modern Earth observation, offering the ability to operate flexibly in complex and hazardous environments, unrestricted by terrain and road networks. They provide high-resolution imagery, enabling decision-makers to rapidly acquire surface information, and have demonstrated significant advantages in flexibility and timeliness in fields such as land surveys [1], disaster emergency response [2], and cultural heritage preservation [3]. However, as the remote sensing monitoring scope expands and tasks complexity increases, a single UAV is often not enough in terms of endurance and operational efficiency.

For instance, during large-scale ground missions such as the 2021 Zhengzhou flood [4] and the 2024 Valencia flood in Spain (https://www.copernicus.eu/sl/node/69707, accessed on 6 August 2025), which affected 328.32 km^2 and 4100 hectares of land, respectively, single-UAV systems are unable to conduct rapid and comprehensive surveys on their own, while Geographic Information Systems (GIS) and advanced remote sensing technologies aid intelligence acquisition in complex environments, collaborative multi-UAV systems are crucial for large-scale, time-sensitive missions, offering enhanced efficiency, robustness, and redundancy [5].

Collaborative multi-UAV systems represent a current focal point of international research [6–9], with many countries having formulated long-term strategies and developed advanced platforms for scientific, military, and commercial applications. Multi-UAV collaboration enables efficient task division and parallel execution, significantly reducing the imagery acquisition window in large or heterogeneous areas while simplifying subsequent data processing complexity. Under the management and control of a central node or integrated center, UAVs can collaboratively detect targets in different regions and share sensor information in real time, forming a network. The collected data are processed and fused across UAVs, enabling resource sharing and mutual support to ensure efficient and seamless coverage of the entire airspace structure. Furthermore, driven by advances in communication, hardware, and intelligent software, the use of UAV networks or swarms has emerged as a key trend in remote sensing research [10,11].

However, the launch, recovery, and coordination of multi-UAV systems require extensive preparatory work, which constrains operational efficiency and substantially increases costs. To enhance deployment flexibility and sustained operation in remote-sensing applications, ground vehicles can be transformed into mobile platforms that integrate UAV transport, rapid deployment, launch and recovery, onboard data processing, and energy replenishment—hereafter referred to as VM-UAV systems [6,12]. These vehicle-mounted platforms not only overcome the transportation, power, and onboard computing bottlenecks faced by small- and medium-sized UAVs during field operations, but also markedly expand the operational envelope of UAVs, showing great potential for time-sensitive tasks such as rapid mapping and disaster response, as shown in Figure 1.

Despite the numerous advantages that vehicle-mounted multi-UAV system bring to remote sensing, developing efficient collaborative scheduling and flight path planning mechanisms to fully exploit their potential remains a major challenge. Most existing UAV ground station software, primarily supports manual route planning for a single UAV, failing to meet the complex demands of multi-UAV collaborative operations [13]. Simply increasing UAV numbers without effective coordination produces uneven task loads and

staggered completion times, so the system's performance is constrained by a "wooden-barrel" bottleneck—the weakest link determines overall capacity. In multi-region operations that require frequent takeoffs and landings, irrational area allocation and suboptimal route design cause excessive non-productive turns, elevating energy consumption and reducing endurance. These inefficiencies delay data acquisition and degrade the timeliness of decision-making, which is critical in emergency response. Consequently, there is a clear gap: current methods do not provide mature, system-level intelligent task allocation and energy-aware path-planning that can fully exploit the capabilities of VM-UAV remote-sensing platforms.

The objective of this research is to develop a practical method that can significantly enhance the operational efficiency, coverage range, and response speed of vehicle-mounted multi-UAV systems in remote sensing surveys, providing robust support for rapid and precise information acquisition in complex environments. The problem is highly challenging as it involves the joint design of multi-UAV scheduling and path planning. First, we design a scheduling scheme that determines the visit order of multiple regions for multiple UAVs to optimize the sequence and minimize inter-region path lengths. Additionally, it balances the tasks of each UAV to reduce the time wasted by other UAVs while waiting. Second, for each location, we implement an optimal coverage flight path that avoids excessive turns, which increase energy consumption, thereby offering flexibility in high-quality coverage and time efficiency based on the UAV type [6,12].

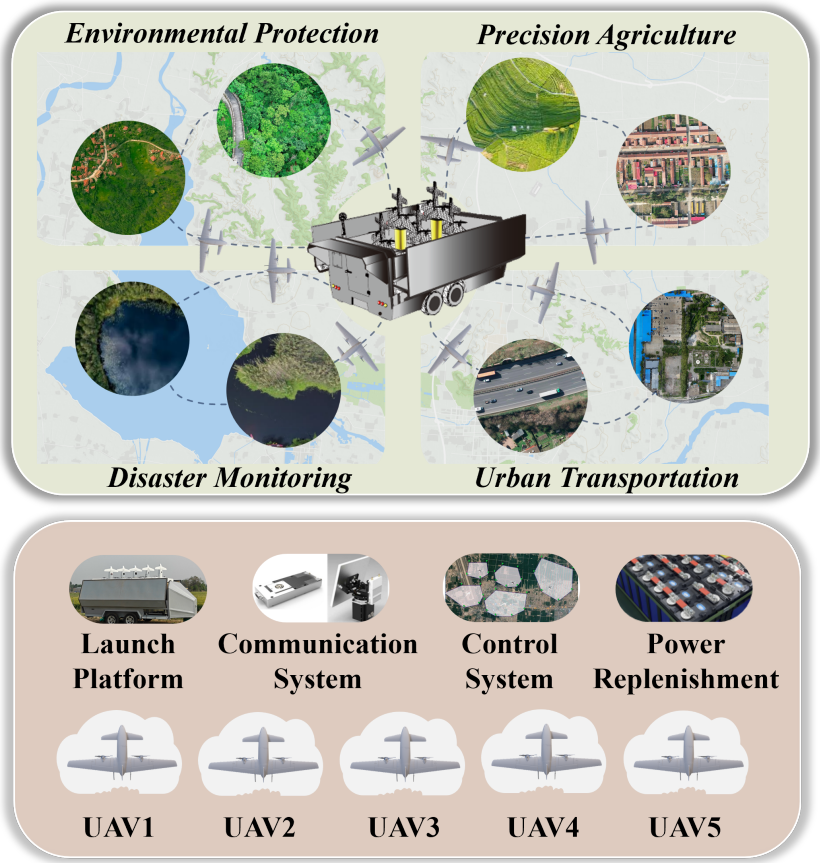


Figure 1. Schematic diagram of a vehicle-assisted multi-UAV remote sensing application.

2. Related Work

Optimizing the scheduling and routing for multi-UAV systems is a fundamental challenge, recognized as an NP-hard problem. It is often modeled as a variant of the Vehicle Routing Problem (VRP) or the Multiple Traveling Salesman Problem (TSP), where a

fleet of energy-constrained UAVs must efficiently visit a set of target nodes. The primary goal is to optimize objectives such as minimizing mission time or energy consumption while respecting constraints like UAV endurance, payload capacity, and flight speed. Therefore, beyond being NP-hard in terms of exact optimization, it is also difficult to approximate within a guaranteed ratio. Consequently, metaheuristic approaches such as genetic algorithms are adopted to efficiently explore the solution space and obtain high-quality approximate solutions in practice [14]. In addition, commonly used methods also include mathematical programming [15] and deep reinforcement learning (DRL).

One stream of research utilizes mathematical programming to formulate exact models, primarily through Mixed Integer Programming (MIP). These models excel at finding optimal solutions for small to medium-sized problems. In the context of logistics and cargo delivery, some models focus on minimizing mission duration by permitting UAVs to be launched and retrieved from different carrier vehicles [16–18]. Other researchers have developed integrated frameworks that simultaneously optimize task scheduling, route planning, and even recharging decisions, thereby managing the overall energy consumption of the mission [19]. Despite their precision, exact methods like MILP and branch-and-bound face significant scalability challenges. Their computation time grows exponentially with problem size, making them impractical for large-scale or time-sensitive scenarios. Collectively, the above studies indicate that carefully designed heuristic approaches can substantially improve the performance and adaptability of UAV task scheduling and routing across diverse environments. In recent years, deep reinforcement learning (DRL) has been applied to UAV path planning to generate complex 3D trajectories and support real-time decision-making. However, DRL for multi-UAV path planning and scheduling still faces several challenges. First, high-dimensional state spaces and continuous action spaces incur substantial computational cost and long training times; second, multi-agent non-stationarity and reward sparsity destabilize learning and worsen sample inefficiency. To mitigate these issues, researchers have adopted hybrid strategies—e.g., offline pretraining to reduce online computation and exploration risk [20], hierarchical task decomposition separating assignment from trajectory control [21], and hybrid model architectures [22] coupled with tailored dense potential-field reward designs to alleviate sparse rewards and accelerate convergence [23]. These methods have produced promising simulation results, primarily for 3D obstacle-avoidance trajectory generation, but challenges remain for scalability, robustness to real-world disturbances, and transfer to physical platforms. To overcome the scalability limitations of exact methods, heuristic and metaheuristic algorithms are widely adopted due to their flexibility and efficiency in finding high-quality, near-optimal solutions for large, complex problems. GA is a popular choice. Early work demonstrated their ability to converge to near-optimal solutions more rapidly than traditional search methods [24]. Subsequent research has focused on enhancing GA with more sophisticated designs, such as dual-chromosome encoding to manage complex task assignments and conflict resolution mechanisms tailored for missions with heterogeneous UAVs [3,12,25,26]. Beyond GAs, alternative approaches include improved PSO and ACO algorithms designed to avoid premature convergence, as well as hybrid methods that first decompose the problem by clustering tasks (e.g., using k-means) and then optimize routes within each cluster [12,27,28].

Beyond scheduling the sequence of areas to visit, a crucial second-level problem is generating an efficient Coverage Path Plan (CPP) for each designated area. The foundational strategy for CPP is often area decomposition, where a complex region is divided into simpler sub-regions that can be covered with systematic patterns, such as the classic boustrophedon (back-and-forth) algorithm [29]. To accommodate real-world irregular environments, researchers have developed advanced algorithms for non-convex polygons.

For instance, Torres et al. introduced techniques like improved polygon decomposition and the "interrupted path" concept, which optimize flight paths in geometrically complex areas, demonstrating excellent performance [30]. In multi-UAV cooperative studies, the decomposition of large operational areas is an important research focus. For instance, a multi-UAV survey of an Adelie penguin colony of approximately 300,000 birds across a 2 km^2 Antarctic area shortened the traditional survey duration from two days to just three hours [31]. Building on this idea, Apostolidis et al. [5] developed a multirotor mission-planning platform that decomposes the survey region using the DRAP method [32] and applies a minimum spanning tree to reduce turns in flight paths. Similarly, Cao et al. [33] combined a sensor-based model with formation-flight coverage-path planning to simulate photogrammetric applications for UAV swarms. These studies illustrate how careful area decomposition and route design can markedly improve the efficiency of large-scale UAV surveys.

However, the above methods considers only independent, isolated survey regions. From a practical standpoint, missions involving multiple regions require both effective allocation and routing of UAVs and the design of coverage strategies. Most current research on coverage task planning focuses on either UAV scheduling or coverage, lacking a unified approach. When large-scale optimization problems arise, new demands are placed on the proposed algorithms. In summary, a new guiding algorithm is needed to provide a comprehensive solution for multi-UAV remote sensing applications.

2.1. Problem Description

Large-scale remote-sensing missions often involve multiple discrete task regions that exceed the number of available UAVs. A vehicle-mounted platform carries M UAVs which operate as a cooperative swarm to perform coverage observations. The objective of this study is to assign each UAV a subset of task regions, each UAV sequentially executes coverage flights over its assigned regions and then returns to the vehicle. The system model assumptions are as follows:

1. Prior to mission execution, the control system has complete knowledge of all task regions, including their number, locations, and boundaries.
2. For remote-sensing mapping, orthophotos are used for wide-area coverage. Thus, each UAV flies at a fixed altitude and constant speed, maintaining constant forward and side overlap rates within each region. The ground sampling distance (GSD) is computed automatically from the flight altitude.
3. UAV energy consumption is modeled as proportional to flight time, and aerodynamic models are not considered. UAVs are assumed to remain undamaged throughout the mission.
4. UAVs are launched from the vehicle and are required to return to the vehicle upon mission completion.

This study tackles two interrelated problems. First, for each task region we seek an energy-optimal coverage path that minimizes in-mission energy expenditure. Second, we develop a task-assignment and sequencing strategy for multiple UAVs that balances workload across UAVs and optimizes the order of region visits to reduce overall mission complexity. The ultimate goal is to minimize the aggregate energy and time costs of the entire multi-UAV system. A schematic of the prototype system is presented in Figure 2.



(a) (b) (c)

Figure 2. (**a**,**b**) Prototype of the vehicle-mounted multi-UAV system. (**c**) Multi-UAV Ground Control Station in flight experiment.

2.2. Scheduling and Coverage Model Formulation

Table 1 list the mission model attributes and parameters for the UAV and camera used in this study. In the formulated model, Equations (1) and (2) represent the sets of UAVs and candidate regions. The final optimized result of the scheduling algorithm is a route for each UAV. Assuming there are N_u UAVs, the set of routes is denoted as in Equation (3):

$$U = U_1, U_2, \ldots, U_i, \ldots, U_{N_u}, \quad i \in 1, \ldots, N_u \tag{1}$$

$$V = P_1, P_2, \ldots, P_i, \ldots, P_{N_r}, \quad i \in 1, \ldots, N_r \tag{2}$$

$$U_i = \{P_{k1}, P_{k2}, \ldots, P_{kn}\}, \quad P_k \in V \tag{3}$$

Table 1. Model parameters and description.

Parameters	Attribute	Description
N_r	Number of Regions	The total number of task regions.
$\mathrm{P}_r = \{(x, y)\}^{N_r}$	Region Location	The geographic centroid of each region.
$S_r = \mathbb{R}$	Region Shape	A sequence of polygons defining the boundary of each region.
N_u	Number of UAVs	The number of UAVs available for mission execution.
$\mathrm{P}_{vs.} = \{(x, y)\}^{N_{vs.}}$	Vehicle Deployment Location	Initial deployment position of the vehicle and UAVs (take off point).
h	Flight Altitude	UAV flight altitude above ground level (m).
o_s	Side Overlap	Lateral overlap percentage between adjacent aerial images.
o_f	Forward Overlap	Longitudinal overlap percentage between consecutive aerial images.
E_u	Endurance Range	Maximum flight range of the UAV (km).
$P_n = (x, y, z)$	Coordinates	Current three dimensional spatial coordinates of the UAV.
ψ	Raw Heading	UAV's heading angle relative to true north.
ϕ	Yaw Angle	UAV's rotation around the vertical axis.
θ	Pitch Angle	UAV's rotation around the lateral axis.
f	Focal Length	Camera lens focal length determining image scale (mm).
v	Cruise Speed	UAV's average cruising speed (m/s).
W	Image Width	Number of pixels in the horizontal direction of the captured image.
H	Image Height	Number of pixels in the vertical direction of the captured image.

The workload of a UAV is determined by the sum of its flight distances, comprising four components: the launch segment, the intra-region coverage paths, the inter-region transit paths, and the return segment, as expressed in Equation (4).

$$L_i = d(P_v, p^{\text{entry}}_{i,j_{i,1}}) + \sum_{k=1}^{m_i} L^{\text{cover}}_{i,j_{i,k}} + \sum_{k=1}^{m_i} d\left(p^{\text{exit}}_{i,j_{i,k}}, p^{\text{entry}}_{i,j_{i,k+1}}\right) + d\left(p^{\text{exit}}_{i,j_{i,m_i}}, P_v\right) \tag{4}$$

where $L^{covs.er}_{i,j_{i,k}}$ represents the total path length required for UAV U_i to complete coverage observation in sub-region j. d denotes the Euclidean distance between two locations, where P^{entry} is the entry point and P^{exit} is the exit point. P_v represents the vehicle's position. The total route cost of the mission, F_{route}, is the normalized sum of all UAV flight ranges, as shown in Equation (5):

$$F_{route} = \frac{\sum_{k=1}^{N_u} L_k}{\sum_{i=1}^{N_u} E_{u,i}} \tag{5}$$

$F_{balance}$ is represented as the variance of the ratio of each UAV's flight path to its maximum endurance, calculated as shown in Equation (6):

$$F_{balance} = \frac{1}{N_u} \sum_{i=1}^{N_u} \left(\frac{L_i}{E_{u,i}} - \frac{1}{N_u} \sum_{j=1}^{N_u} \frac{L_i}{E_{u,i}} \right)^2 \tag{6}$$

The minimization objective function is the weighted sum of path length and load balance, where σ are weight parameters, as

$$F_{cost} = \sigma_r F_{route} + \sigma_b F_{balance} \tag{7}$$

where σ_r is the route cost weight parameter, and σ_b is the load balance weight parameter. In this study, the route cost weight is set proportional to the sum of the ratios of each UAV's flight distance to its maximum endurance, while the load balance weight is computed using a normalization-based method. As the number of UAVs increases, the route cost weight rises significantly. Therefore, in all algorithms, the route cost weight is set to 1, and the load balance weight is set equal to the number of UAVs. In practical applications, these two parameters can be adjusted according to mission priorities to flexibly achieve different operational objectives.

2.3. Camera Model

The objective of UAV remote sensing is to acquire complete imagery that covers the target area while satisfying resolution requirements. During image acquisition, the overlap between consecutive frames together with flight altitude (GSD) jointly govern the quality of the reconstruction. The overlap patterns and image footprints are illustrated in Figure 3.

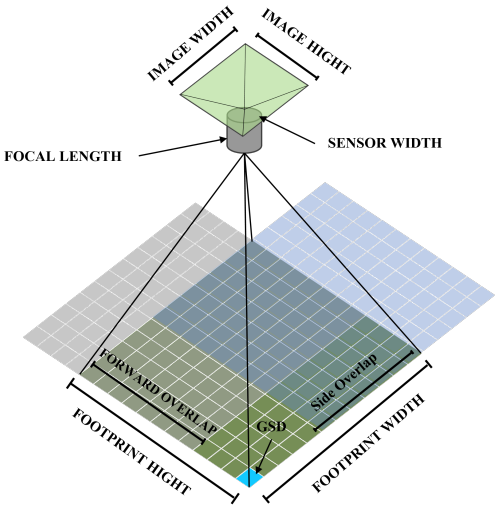


Figure 3. Flight line overlap and footprint in UAV photogrammetry.

The ground sampling distance (GSD) refers to the distance between the centers of two adjacent pixels on the ground and can be automatically calculated based on the sensor width S_w using Equation (8).

$$GSD = 100 \frac{hS_w}{Wf} \tag{8}$$

The ground footprint of a single image can be derived from the UAV' position $\mathbf{P}_n$ and pose $R_{c\to W} = R_z(\psi)R_y(\theta)R_x(\phi)$. The footprint projected on the terrain has a length l and width w. The forward overlap and side overlap rates are calculated as in Equations (9) and (10):

$$o_l = 1 - \frac{R_d}{w \cdot GSD} \tag{9}$$

$$o_s = 1 - \frac{sh \cdot v}{l \cdot GSD} \tag{10}$$

where R_d is the distance between adjacent flight lines, sh is the shutter interval between two aerial images.

3. Energy-Optimal Coverage Planning Model

To alter its heading during flight, a UAV must decelerate and adjust its attitude, requiring additional thrust to counteract centrifugal forces and safely negotiate a turn. This is particularly critical for fixed-wing UAVs, whose minimum turning radius is a function of velocity, lift coefficient, and load factor. Consequently, minimizing the frequency of high-curvature turns within a coverage path plan is essential for reducing overall energy expenditure. Neglecting the significant energy consumed during these maneuvers can lead to a critical underestimation of the total power budget, potentially resulting in insufficient flight endurance, a premature return-to-launch, or an emergency landing. Therefore, the current coverage planning path needs to go beyond geometrical "full coverage" and seek to maximize time, energy, and cost efficiency while meeting physical kinematic constraints [34].

To solve the single-region coverage problem, we reference the work of Krupke [34] to achieve an approximately optimal coverage path with turning costs. The method primarily involves the following five steps: area discretization and value estimation, linear relaxation solving, atomic strips selection, constructing a cycle cover, and path connection and optimization.

3.1. Area Discretization and Value Estimation

First, the complex polygonal area to be covered is discretized into a graph $G = (P, E)$ through equilateral triangulation, yielding a set of points and edges. For each node $v \in P$, its Voronoi cell on the grid is constructed, and the area of this cell is taken as the node's coverage value, $val(v)$, reflecting the approximate covered area contributed if the path covers this node. The distance between adjacent nodes in the grid is set to $1.15R_d$ to maintain the side overlap rate of the aerial images.

3.2. Linear Relaxation Solving

On the graph G a three-way path variable χ_{uvw} is introduced to represent the frequency of the path $u \to v \to w$. A skip variable s_v is also introduced, where skipping a node incurs a penalty equal to $val(v) \cdot s_v$. Linear programming is used to optimize the objective function on graph G :

$$min \sum_{v \in P} val(v) \cdot s_v + \sum_{(u,v,w)} cost(u, v, w) \cdot x_{uvw} \tag{11}$$

Here, the cost function $cost(u, v, w)$ incorporates the Euclidean distance and a penalty for turning angles, and local weights α may be assigned to nodes. The constraints comprise a "cover-or-skip" constraint for each node v:

$$\sum_{u,w} x_{uvs.w} + s_{vs.} \geq 1 \tag{12}$$

which ensures that every node is either covered by a path or incurs its skip cost, and a flow-conservation constraint that enforces balance of the fractional flow on each undirected edge at both endpoints. Solving this linear program produces a fractional solution $\{x^*_{uvw}, s^*_v\}$, which provides a lower bound and guides the construction of a feasible integer solution.

3.3. Atomic Strips Selection

The fractional solution needs to be converted into a set of atomic strips for matching. Each atomic strips represents a small hypothetical movement near a node v with a fixed orientation. This allows the turning cost to be approximately mapped to matching edges. For each node v, all its adjacent edges are enumerated as candidate orientations. The "fractional flow alignment" is calculated for each orientation, which is the sum x^*_{uvw} of for all path (u, v, w) passing through v where the direction $w \to v$ is closest to the orientation. The best few orientations for each node are retained as candidate atomic strips. For each selected atomic strips ζ corresponding "endpoint nodes" are created at its ends. A matching edge is established between any two endpoints. If the two endpoints come from adjacent nodes with a consistent geometric orientation, the edge weight is the half-segment distance and turning cost. If they are the two ends of the same atomic segment, the matching edge weight is set to the penalty for skipping that node $val(v)$. This results in an endpoint matching graph $\widetilde{G}$ with approximately $O(|P| \cdot k)$ vertices and $O(|P| \cdot k2)$ edges.

3.4. Constructing Cycle Cover

In the endpoint matching graph $\widetilde{G}$, the Blossom V [35] algorithm is used to find a minimum weight perfect matching, which results in a set of pairs (a_i, b_i). Each pair (a, b) represents either the connection of two atomic segment endpoints to form a path fragment or the skipping of the node and paying its value penalty. The matching results are projected back onto the original graph which G, splices together several disjoint cycle covers. These cycles globally balance coverage value and turning costs while avoiding excessive backtracking. Figure 4 illustrates the atomic strips selection and the Cycle Cover.

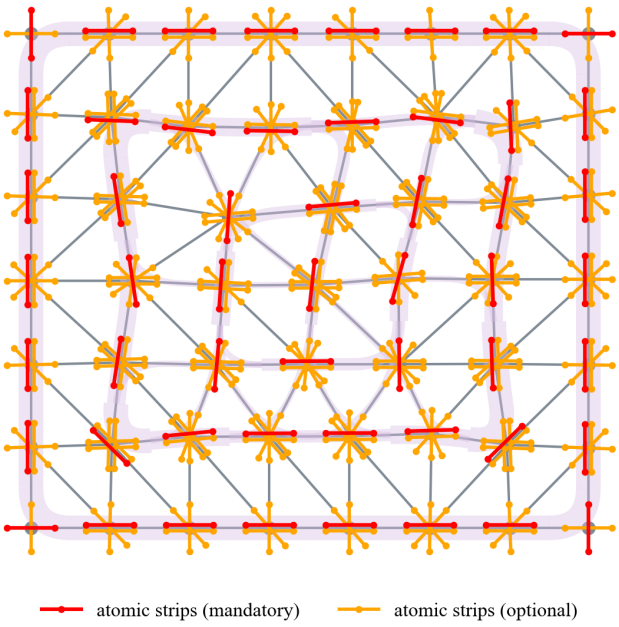


Figure 4. Example of atomic strips selection. Atomic strips are shown in yellow (optional) and red (mandatory). The matching result is displayed in light purple.

3.5. Path Connection and Optimization

The initially generated paths are optimized using local MIP. A highest cost node v_{root} is selected, and a Breadth-First Search is performed from it to construct a subgraph G_{loc} containing the first t nodes. In this subgraph, the corresponding path and skip variables are set as binary, while external variables are fixed to the ceiling of the fractional solution. After solving for the subgraph, if the objective value decreases and the local solution can connect to the external solution, the global solution is updated. This iterative process reduces redundant turns and skips unnecessary nodes. The Prize-Collecting Steiner Tree (PCST) model is introduced, where the edge weight between two cycles is their shortest connection cost. Solving the PCST yields a "value-maximized" connected subtree. Only the cycles selected in the PCST are merged, while the rest are discarded. Geometrically adjacent cycles with low connection costs are merged into a larger cycle using the Dijkstra algorithm to compute the shortest path with turning costs. Finally, a global local MIP optimization is performed on the connected cycle to further eliminate redundancies and overlapping coverage introduced during merging.

4. Multi-UAV Scheduling and Routing

GA have proven to have scalability and speed advantages in solving large-scale discrete solution spaces. However, traditional GA often struggle to find the optimal solution for multi-UAV task scheduling and routing [13]. Therefore, improvements are needed in three areas: encoding and decoding, crossover operator, and mutation operator. In this section, we introduce the framework of our proposed improved genetic algorithm, its encoding method, new crossover operator, and the adaptive hybrid mutation strategy. The system framework is illustrated in Figure 5.

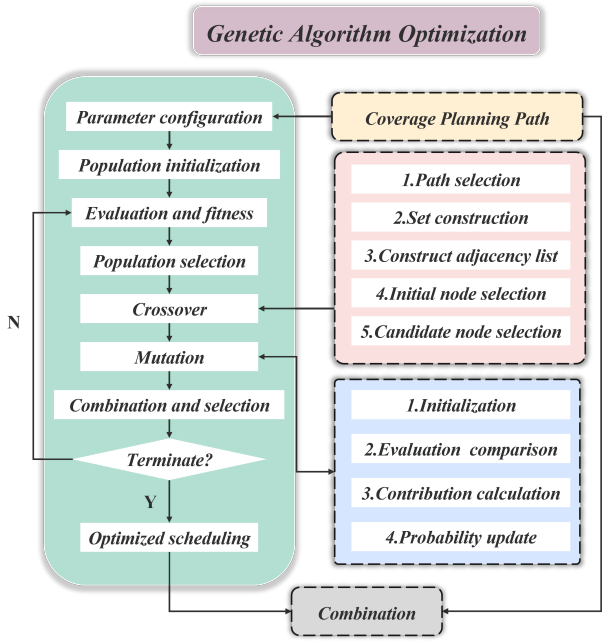


Figure 5. Framework of the GA for multi-UAV routing and scheduling.

The steps of the proposed algorithm are as follows:

Step 1. Model Initialization and Coverage Path Construction. Calculate the flight altitude and flight line width based on mapping requirements. For each region a UAV needs to visit, first plan the coverage path cost as a weight for the region's location, as described in Section 4.

Step 2. Population Initialization. The initial population of individuals is created by randomly shuffling the regions and assigning them to UAVs, ensuring the diversity of the initial population.

Step 3. Objective Function and Fitness Evaluation. The fitness of individuals' path connection schemes is calculated according to the objective functions for route cost and load balance. The framework adopts a multi-objective optimization strategy that aims to minimize the flight distance and optimize the load balance of each UAV.

Step 4. Selection. As a multi-objective optimization problem, NSGA-II is used as the basic framework. It selects individuals through non-dominated sorting and crowding distance to maintain population diversity and converge to the Pareto front.

Step 5. Crossover. Use the Multi-Region Edge Recombination Crossover operator from Section 4.3.

Step 6. Mutation. Use the Adaptive Hybrid Mutation Mechanism from Section 4.4.

Step 7. Merge Population. The chromosomes with the highest fitness are selected to join the offspring population. In each generation, a new population of μ individuals are selected from the μ parent individuals and λ offspring individuals. This means both parents and offspring have a chance to enter the next generation, which is a form of elitism that helps preserve excellent individuals.

Step 8. Termination. If the number of iterations is met, stop the loop and generate the task allocation and path planning results; otherwise, return to Step 2.

Step 9. Combination. The coverage paths for all regions and the transitions between them are connected using Dubins curves under flight radius constraints. Connect all regions and the vehicle to form the final paths.

4.1. Chromosome Encoding

To adapt to the multi-UAV routing and scheduling problem, Wang et al. [25] used a dual-chromosome encoding, but their method of cutting based on position could lead to a large number of infeasible solutions. J. Li et al. [13] improved the dual-chromosome encoding by reserving unallocated space for each UAV's tasks, but this fixed the number of regions each UAV could visit, requiring pre-mission configuration. To avoid this issue, this study modifies the chromosome encoding method. An individual is defined by N_U routes, where each route $I = (R_1, R_2, \ldots, R_d)$ is a permutation of task nodes to be visited. Each route starts from launching platform, visits all nodes in its list, and then returns to the vehicle. The chromosome uses integer encoding, as shown in Figure 6. Layer I represents the queue of tasks to be performed, and Layer II represents the UAV assigned to perform those tasks. In Layer I, each UAV is assigned a list, with the vehicle index at the head and tail. After task initialization, the middle of the list is filled with the region visit tasks. The vehicle index always remains at the beginning and end of each UAV's assigned task list and does not change its position. Thus, the first layer of genes is represented by N_U lists, containing N_r non-repeating regions. The second layer is the index for the UAV ID. The advantage of this improved chromosome encoding is that it removes the upper limit on the number of tasks, reduces the complexity of model initialization and the need for manual intervention, allows for a higher optimization ceiling in highly heterogeneous tasks, and facilitates task exchange and matching based on constraints.

Layer1	[0,1,4,5,7,0]	[0,13,16,2,8,0]	[0,9,11,6,12,0]	[0,3,10,14,15,0]
Layer2	UAV1	UAV2	UAV3	UAV4

Figure 6. Example of chromosome encoding.

Figure 6 illustrates an example of the improved two-layer chromosome encoding scheme, designed for a scenario involving 16 regions and 4 UAVs. The first layer specifies the region visiting sequences for the four UAVs, which are given as: [0, 1, 4, 5, 7, 0], [0, 13, 16, 2, 8, 0], [0, 9, 11, 6, 12, 0], and [0, 3, 10, 14, 15, 0]. Here, the number "0" denotes the depot location of the ground vehicle. Taking UAV 1 as an example, after deployment from the vehicle, it sequentially visits regions with index IDs 1, 4, 5, and 7 in the specified order to accomplish its coverage task.

4.2. Population Selection

The genetic algorithm employs the NSGA-II framework. During the selection process, the dominance relationship is determined by Pareto dominance, which stratifies the population into non-dominated fronts. A solution x is said to dominate a solution y if

$$\begin{cases} f_1(x) \le f_1(y) \wedge f_2(x) \le f_2(y) \\ \exists j \in 1,2\colon f_j(x) < f_j(y) \end{cases} \tag{13}$$

where f_1 is the path cost F_{cost} and f_2 is the load balance $F_{balance}$. Within the same front, the crowding distance I_d is calculated to measure the density of solutions. The crowding distance of solution i is given by

$$I_d(i) = \sum_{m=1}^{2} \frac{f_m(i+1) - f_m(i-1)}{f_m^{max} - f_m^{min}} \tag{14}$$

During the crowding distance calculation, the distance for boundary solutions is set to infinity to ensure a uniform distribution of solutions in the objective space. Finally, binary tournament selection is used to choose parents. Solutions with a lower front rank are preferred. If two solutions are in the same front, the one with the larger crowding distance is chosen:

$$\begin{cases} r_a < r_b, choose\ x_a \\ \{r_a = r_b\ \Lambda\ I_d(a) < I_d(b), choose\ x_a \end{cases} \tag{15}$$

The NSGA-II framework preserves high-quality solutions while maintaining population diversity through crowding distance, ultimately generating a balanced set of solutions that cover the Pareto front.

4.3. Multi-Region Edge Recombination Crossover

In multi-path optimization problems like multi-UAV path planning, preserving the superior edge structures of parent solutions is crucial for generating high-quality offspring. The Edge Recombination Crossover (ERX) operator [36] inherits the connectivity relationships between elements from parent individuals by constructing an adjacency list, producing offspring that retain path continuity. Inspired by the classic ERX, we introduce MRECX to implement a path-optimizing crossover operator.

The MRECX operator is applied with a probability p_c to create a pair of offspring chromosomes O_A and O_B from a pair of parent chromosomes P_A and P_B. This crossover only applies to the UAV's regional visit paths. If crossover is performed, for each corresponding UAV index $j \in [1, N_U]$, MRECX extracts the parent routes $R_{(A,j)}$ and $R_{(B,j)}$, determines the unique set of task nodes to be visited by U_j as

$$U_j = Set(R_{(A,j)}) \cup Set(R_{(B,j)}) \tag{16}$$

An adjacency list is constructed for each node v in $U_j \cup P_v$. The edges from both $R_{(A,j)}$ and $R_{(B,j)}$ are added to this list. Each entry records the adjacent node ID, the distance, and a flag indicating if it is a common edge present in both parent routes.

To generate the offspring route $O_{(A,j)}$, it is initialized as empty. A starting node $s \in U_j$ is randomly selected, added to $O_{(A,j)}$ and marked as visited. Let the current node be $curr = s$, while the length of $O_{(A,j)}$ is less than the required length, the following steps are repeated to select the next node *next*:

1. Identify all neighbors of *curr* from the adjacency list that are in U_j and have not yet been visited.
2. If there are candidate neighbors, select next based on the following priority:
   (a) Prefer a neighbor that forms a common edge with *curr*.
   (b) If there are multiple common-edge neighbors or none, prefer the neighbor connected by the shortest edge distance.
   (c) If ties still exist, prefer the neighbor that has the fewest unvisited neighbors itself. This heuristic reduces the risk of isolating other nodes later.
3. If there are no such neighbors (i.e., *curr* is an isolated node), randomly select an unvisited node from the remainder of U_j to be next.
4. Add next to $O_{(A,j)}$ mark it as visited, and update $curr = next$.

This process continues until $O_{(A,j)}$ is a complete permutation of the same length as the parent routes. The same procedure is applied to create $O_{(B,j)}$. Finally, all individual routes are combined to form the complete offspring individuals $O_{(A,j)}$ and $O_{(B,j)}$. Figure 7 shows an example of MRECX application.

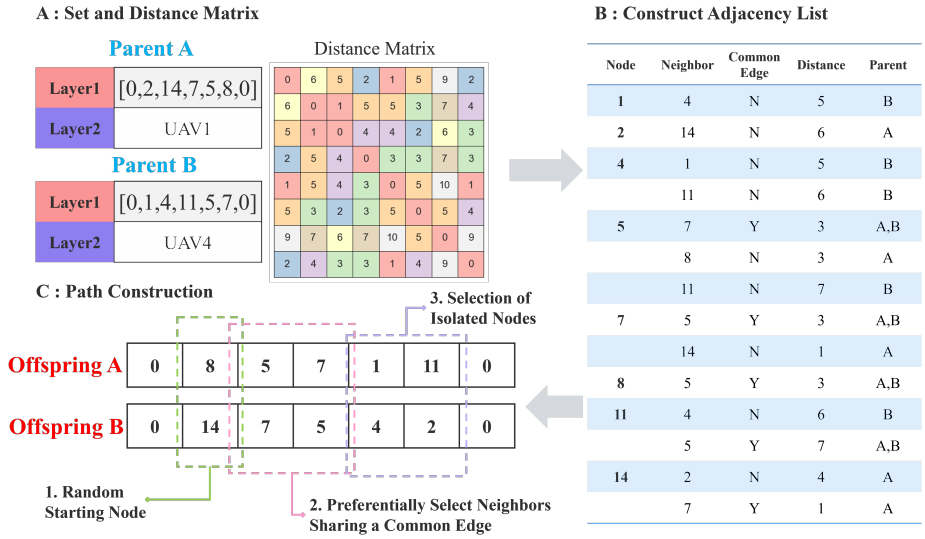


Figure 7. Example of MRECX application.

MRECX preserves common edges from parent chromosomes during iteration, maintaining superior path segments and guiding the evolution in a reasonable direction. Simultaneously, the distance-based selection for isolated nodes helps accelerate convergence and avoid excessive randomness.

4.4. Adaptive Hybrid Mutation Mechanism

The AHM mechanism applies mutation to selected individuals in each generation with an overall mutation probability p_m. When triggered, the operation is performed on the individual. Its core components include a portfolio of mutation operators, an operator selection and application module, a contribution calculation module, and a probability self-adaptation module.

AHM maintains a portfolio $M = \{M_1, M_2, \ldots, M_{N_{OP}}\}$ of N_{OP} distinct mutation operators. Each operator M_i has a specific perturbation mechanism designed to explore the solution space from different perspectives. The portfolio in this study includes the following primary operators: Inversion Mutation, Scramble Mutation, Displacement Mutation, and Task Migration Mutation, as shown in Figure 8.

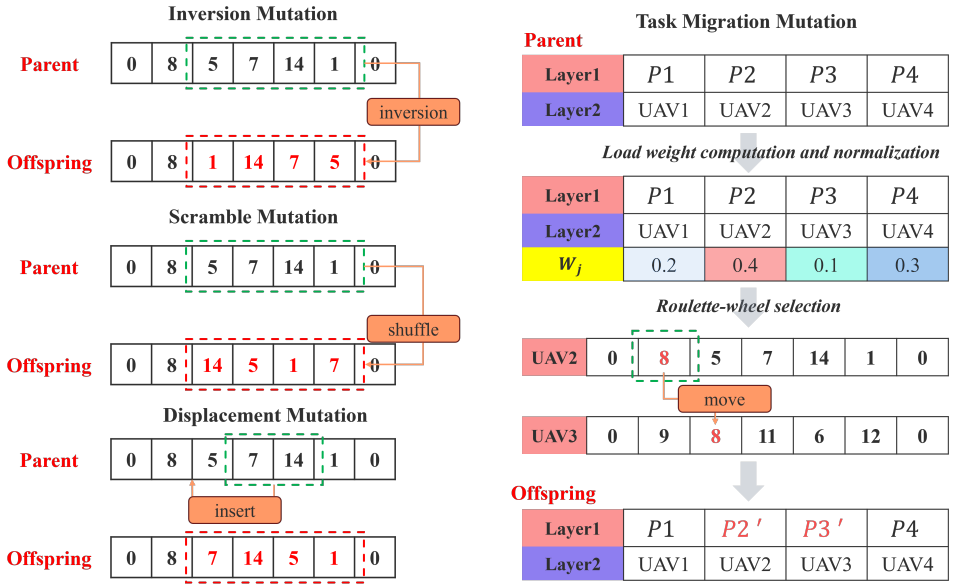


Figure 8. Example of mutation operators.

Inversion Mutation: This operator acts on each route R_j of an individual with an independent sub-probability p_{sub}. If the route length is greater than two, it randomly selects a sub-segment $R_j[a \ldots b-1]$ and reverses the order of its elements. This operator explores neighboring solutions by altering the local task sequence.

Scramble Mutation: Furthermore, applied to each route R_j with probability p_{sub}. If the route length exceeds two, it selects a random sub-segment $R_j[a \ldots b-1]$ and completely shuffles the order of its elements. This operator introduces a greater degree of perturbation than inversion mutation.

Displacement Mutation: Applied to each route R_j with probability p_{sub}. For routes longer than two, it randomly selects a sub-segment $R_j[a \ldots b-1]$, removes it from its original position, and inserts it into a new random position within the remaining part of the route. This operator can significantly alter the global position of a block of tasks.

Task Migration Mutation: If selected, this operator is executed once on the entire individual (all routes). Its purpose is to reallocate tasks between different UAVs to achieve better load balance or cost-effectiveness. The process is as follows:

1. For each route R_j, load weight W_j is calculated, which synthesizes the total intrinsic weight of tasks in the route and its expected execution cost, and is then normalized.
2. Source route R_s is selected from all routes using roulette wheel selection based on the load weights W_j. Routes with higher loads have a greater chance of being selected.
3. The route with the current minimum load weight W_k is chosen as the target route R_T, ensuring $R_s \neq R_T$.
4. If both R_s and R_T contain a sufficient number of tasks (at least two), a region $\tilde{r}$ is randomly removed from R_s.
5. The region $\tilde{r}$ is then added to R_T. The insertion position is determined by a greedy best-insertion strategy, which tries all possible insertion points and selects the one that results in the minimum increase in the estimated cost of R_T.

When mutating an individual, an empty applied operator set S_{apply} is initialized. For each operator M_i, a random probability $rand \in [0,1]$ is generated. If $rand \leq P(M_i)$, then M_i is added to S_{apply}. $P(M_i)$ is the current dynamic application probability of operator M_i, which is initialized equally for all operators. If is still empty after this probabilistic selection, one operator is chosen uniformly at random from M to ensure at least one mutation is performed. All operators in S_{apply} are then applied sequentially to a copy of the individual. After applying the operators, the fitness of the mutated individual f_{mutant} is evaluated and compared to the original individual's fitness f_{raw} to calculate a composite improvement score I_{mp}. For a multi-objective minimization problem, I_{mp} is calculated as

$$Imp = \sum_{i=1}^{2} \frac{max\left(0, f_{raw,obj_i} - f_{mutant,obj_i}\right)}{max\left(\epsilon, f_{raw,obj_i}\right)} \tag{17}$$

where ϵ is a small positive number to prevent division by zero. This score I_{mp} is then evenly distributed among all operators in S_{apply}. For each $M_i \in S_{apply}$, its contribution for this application is recorded as $Contrib_i = I_{mp}/|S_{apply}|$. If $I_{mp} > 0$, the success counter for each $M_i \in S_{apply}$ is incremented by $1/|S_{apply}|$.

The application probabilities $P(M_i)$ for each mutation operator are periodically adjusted every G_{update} generations. First, for each operator M_i, its average contribution $\bar{C}_i$ over its last Wapplications is calculated. A new raw probability P'_{raw} is then computed based on this average contribution:

$$P'_{raw}(M_i) = \frac{\bar{C}_i}{\sum_{l=1}^{N_{op}} \bar{C}_l} \tag{18}$$

To ensure all operators have an opportunity to be explored, a minimum probability P_{min} is set. The adjusted probability is $P'_{adj}(M_i) = max(P_{min}, P'_{raw}(M_i))$. Finally, all adjusted probabilities are normalized to obtain the new application probabilities:

$$P(M_i) = \frac{P'_{adj}(M_i)}{\sum_{l=1}^{N_{op}} P'_{adj}(M_l)} \tag{19}$$

5. Experimental Results and Analysis

In this section, we analyze and discuss the optimization capability of different algorithms for solving the multi-UAV scheduling and routing problem. To rigorously evaluate the benefits of our approach, we implemented a baseline genetic algorithm (GA-Default) within the proposed framework that employs Partially mapped crossover (PMX) [37] and task-migration mutation operator. To demonstrate the effectiveness of the individual modules, we replaced the PMX operator with the proposed MRECX crossover to obtain GA-MRECX. Finally, by augmenting GA-MRECX with the AHM module, we formed the complete method used in this study (GA-Proposed). Drawing on prior work, we also implemented multi-UAV scheduling algorithms based on PSO [38], ACO [39], and the spatial clustering–traveling salesman algorithm (SC-TSA) [12].

To evaluate the performance and applicability of the proposed algorithm, we conducted numerical simulations and tested it on two typical study areas. Table 2 lists the settings for the numerical simulations, detailing parameters for four scenarios of varying scale. The Route Weight is uniformly set to 1, and the Balance Weight is set equal to the number of UAVs to emphasize the need for energy balancing in multi-UAV environments. Four scenario datasets of varying scale were generated randomly to evaluate the method's optimization performance and generalizability. Representative study areas for the algorithm include wildfire-affected Maui, Hawaii (USA), and Zhoukou, Henan (China), which

is characterized by dispersed rural residential settlements. All experiments were performed on a mobile computer with a 13th Gen Intel(R) Core(TM) i5-13500HX 2.50 GHz processor and 8 GB of RAM. All algorithms were implemented in Python 3.12. The parameters of the algorithm settings are shown in the Appendix A.

Table 2. Simulation parameters for the datasets.

Datasets	I	II	III	IV
UAVs	4	8	12	16
Regions	30	50	70	100
Area	4 km^2	16 km^2	36 km^2	64 km^2
Complexity	0.9	0.8	0.7	0.6
Coverage Cost	1∼10 km	1∼10 km	1∼20 km	1∼20 km
Endurance	50 km	70 km	70 km	100 km
σ_r	1	1	1	1
σ_b	4	8	12	16

Complexity represents the randomness of the location distribution and the complexity of the region shapes. The specific implementation refers to Krupke's work [34].

5.1. Numerical Simulation

The numerical simulation results are shown in Table 3. In all models, the GA-Proposed in this study achieved the best results across all simulated scenarios, with F_{cost} values of 3.435, 5.215, 7.186, and 8.064 for the four datasets, respectively. Compared to other algorithms, GA-Proposed achieved a better balance between route cost and load balancing, indicating more reasonable task allocation and scheduling. GA-MRECX exhibits a considerable improvement over GA-Default, demonstrating the superiority of the chromosome encoding scheme and the MRECX operator proposed in this study. The next best performing algorithms were ACO, PSO, and GA-Default. The distinguishing feature of the SC-TSA method is that it decouples task scheduling from path planning. During the spatial clustering process, routing tasks are aggregated into more compact regions, which facilitates construction of reasonable single-UAV path costs; however, because the method does not simultaneously account for multi-UAV load balancing, the overall F_{cost} cannot be reduced to a low level. In contrast, the other heuristic algorithms incorporated task migration mutation or similar strategies during iteration, enabling them to achieve more balanced task assignments. In terms of time efficiency, SC-TSA is very efficient on smaller datasets, whereas PSO becomes more efficient as dataset size grows. The proposed method requires tracking the mutation operator's application probability during iterations, which reduces its time efficiency. Nevertheless, it can generate paths within on the order of tens of seconds, which is acceptable for many practical scenarios.

Figure 9 shows the iterative optimization process in four task scenarios to compare the solution performance of the algorithms, while GA-MRECX and GA-Proposed exhibit nearly identical convergence trends, GA-Proposed explores a broader solution space, demonstrating the effectiveness of the AHM mechanism. ACO and PSO are widely used in combinatorial optimization, they have inherent and practical limitations in multi-UAV multi-path optimization tasks that prevent them from reaching a global optimum. ACO relies on pheromone concentration on local paths to guide its search, which can lead to premature convergence on suboptimal routes. Pheromone updates lag, providing slow feedback on newly discovered high-quality sub-paths, making it difficult to maintain global search vitality. Thus, it converges early but fails to explore the solution space deeply. PSO, originally designed for continuous spaces, must simulate velocity updates for discrete paths through swaps or priorities, lacking theoretical convergence guarantees. PSO can converge quickly, but it often suffers from a loss of diversity, making it nearly

impossible to escape from clusters of homogeneous solutions. The GA-Default, using the PMX operator, preserves absolute positional information rather than adjacency information. This mechanism frequently disrupts beneficial sub-tours (or favorable adjacencies) from the parents, leading to offspring with low structural resemblance. This results in highly random offspring and limited optimization capability, making it difficult to find the optimal solution.

Table 3. Simulation results for the optimization algorithms.

Datasets	Model	F_{route}	$F_{balance}$	F_{cost}	T_s
I	GA-Default	3.493	0.011	3.539	1.858
	GA-MRECX	3.432	0.011	3.476	2.10
	GA-Proposed	3.395	0.01	**3.435**	3.10
	PSO	3.542	**0.009**	3.577	1.807
	ACO	3.452	0.013	3.502	6.186
	SC-TSA	**3.37**	0.71	6.21	**1.565**
II	GA-Default	5.347	0.015	5.466	2.326
	GA-MRECX	5.129	0.013	5.233	3.915
	GA-Proposed	5.07	0.018	**5.215**	4.38
	PSO	5.357	**0.011**	5.443	2.137
	ACO	5.229	0.019	5.38	10.026
	SC-TSA	**4.915**	0.43	8.355	**2.1970**
III	GA-Default	7.928	0.022	8.191	4.47
	GA-MRECX	7.37	0.021	7.622	5.349
	GA-Proposed	6.983	**0.017**	**7.186**	6.868
	PSO	7.975	0.031	8.346	3.533
	ACO	7.384	0.037	7.826	12.423
	SC-TSA	**6.762**	0.261	9.892	**3.457**
IV	GA-Default	8.521	0.038	9.126	4.735
	GA-MRECX	8.017	0.015	8.257	6.664
	GA-Proposed	7.949	**0.007**	**8.064**	8.216
	PSO	8.504	0.023	8.875	**4.253**
	ACO	8.074	0.03	8.548	14.671
	SC-TSA	**7.062**	0.201	10.273	4.437

The bold font in the table footer indicates the best results in the dataset.

Our proposed method uses MRECX to preferentially select common and high-quality edges, aiming to pass validated superior path segments from parents to offspring, thereby potentially accelerating convergence towards high-quality solution spaces. At the same time, it maintains population diversity through heuristic rules (such as selecting the node with the fewest subsequent neighbors) and random selection (for handling isolated nodes). Furthermore, the proposed AHM strategy integrates multiple mutation operators with different search characteristics and combines them with a dynamic probability adjustment mechanism based on historical performance. This enhances the robustness and efficiency of the genetic algorithm when solving complex multi-path optimization problems. The strategy can automatically favor more effective mutation methods based on feedback during the evolutionary process, guiding the search to converge faster to high-quality solutions and better adapt to different problem instances or search phases. Consequently, the proposed method demonstrates significantly better convergence and optimization capabilities, and the diversity of the mutation strategy allows the algorithm to continuously update and seek the optimal solution during iteration.

To accurately assess the performance and robustness of the different algorithms in an uncertain environment, we employed Monte Carlo simulations, conducting 40 repeated

experiments for each numerical datasets. This allowed for a statistically significant in-depth analysis of each algorithm's effectiveness, stability, and relative advantages, ensuring the objectivity and generality of the comparison results. As shown in Figure 10, the proposed method exhibited excellent stability. The box plot is very compact, indicating minimal fluctuation in the results across repeated experiments, with results being highly concentrated except for a few minor outliers. In contrast, GA-Default showed greater volatility and significant error, indicating a lack of robustness to environmental changes, likely due to the strong randomness of the PMX crossover operator, while GA-MRECX shows a considerable improvement over GA-Default, confirming the efficacy of MRECX, its performance ceiling remains lower than that of GA-Proposed. PSO's median was close to GA-Default's, but its box was narrower, suggesting relatively better stability. Although ACO generally performs well with relatively stable results, its cost is approximately 0.3–0.8 units higher than that of GA-Proposed. SC-TSA's results were concentrated at a high-cost level, far above the other metaheuristic methods, but were extremely stable with very little variation.

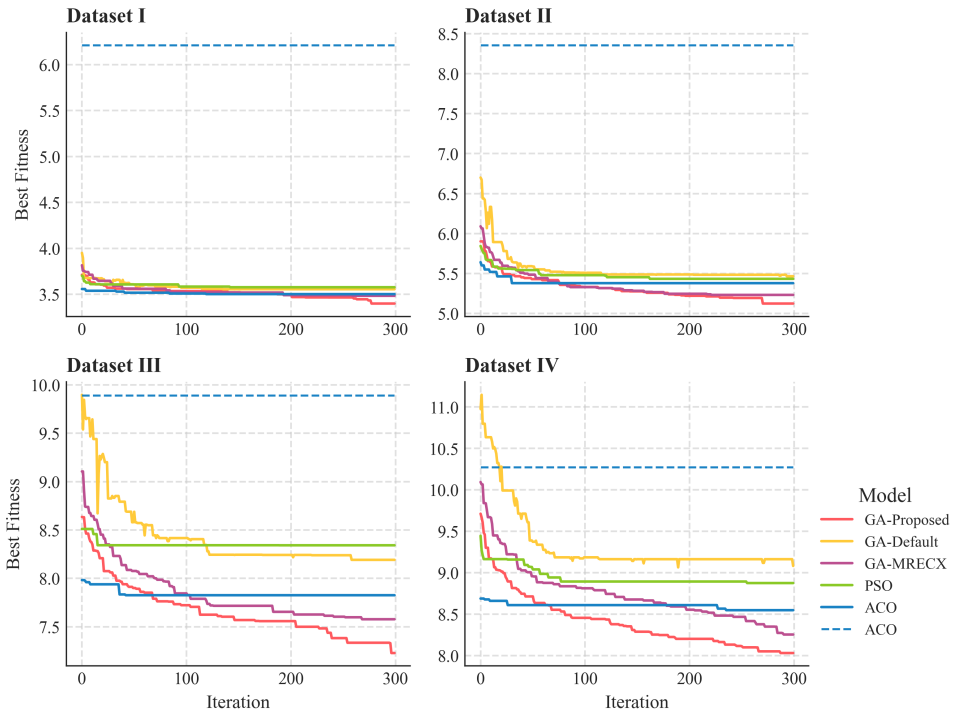


Figure 9. Iteration process of the different optimization algorithms on the four datasets.

To validate the significance of the algorithmic differences, a non-parametric statistical analysis was conducted on the Monte Carlo simulation results. A Kruskal–Wallis H test [40] was performed independently for each dataset, followed by Dunn's post hoc test with Bonferroni correction. The detailed results are presented in Table 4.

The analysis indicated that the performance distributions of the six algorithms exhibited statistically significant differences across all datasets, exemplified by the result for Dataset IV ($H = 226.27$, $p < 0.001$). The post hoc analysis further showed that GA-Proposed performed statistically significantly better than GA-Default, ACO, PSO, and SC-TSA across all four datasets (all $p < 0.0125$). However, under this stringent correction, the performance difference between GA-Proposed and GA-MRECX (all $p > 0.149$) was not found to be statistically significant. This suggests that while GA-Proposed robustly outperforms classical metaheuristics and baseline methods, its performance is statistically comparable to GA-MRECX in these test scenarios.

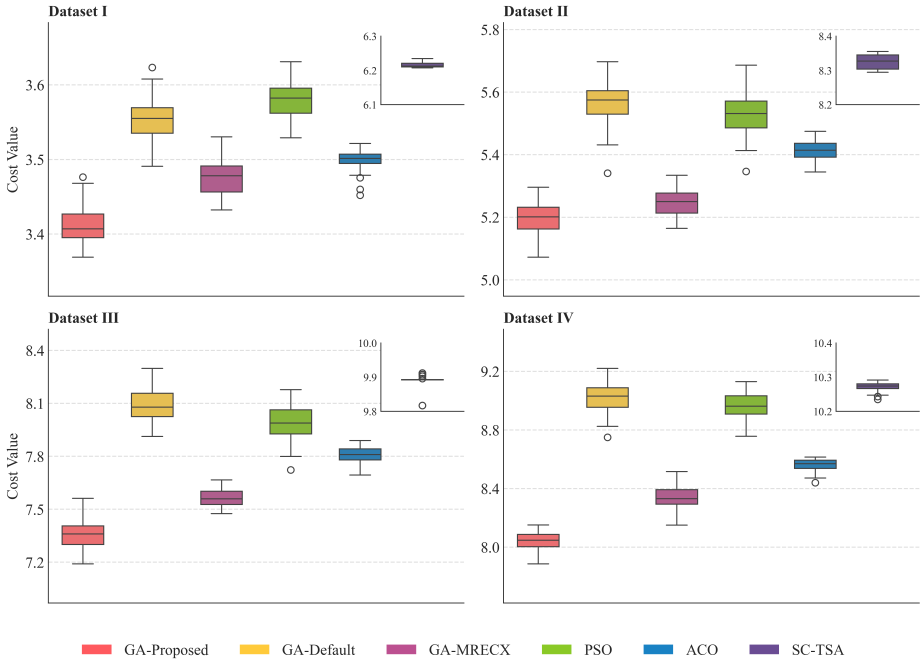


Figure 10. Box plot of the Monte Carlo simulation results.

Table 4. Kruskal–Wallis test results and post hoc significance of GA-Proposed versus baseline methods.

Dataset	*H*	*p*-Value	Significant	Better Than
I	219.3347	2.06×10^{-45}	Yes	ACO, GA-Default, PSO, SC-TSA
II	217.8611	4.27×10^{-45}	Yes	ACO, GA-Default, PSO, SC-TSA
III	224.9819	1.27×10^{-46}	Yes	ACO, GA-Default, PSO, SC-TSA
IV	226.1704	7.08×10^{-47}	Yes	ACO, GA-Default, PSO, SC-TSA

KW: Kruskal–Wallis test. "Significant" denotes rejection of the null hypothesis at $\alpha = 0.05$. Post-hoc comparisons were used to determine which methods GA-Proposed outperformed.

Through numerous repeated experiments and by synthesizing the results from Table 2 and Figure 10, it can be observed that in smaller scenarios, the optimization capabilities of the heuristic-based methods tend to be similar, all finding near-optimal solutions. As the scale of the scenario and the number of UAVs increase, the differences between the methods become more pronounced. Our method maintained the optimal cost planning in all scenarios, indicating that the proposed approach is more likely to converge to a near-optimal solution for large-scale, coupled multi-UAV scheduling problems.

To validate the selection of mutation operators, an ablation study was conducted by removing one mutation operator at a time from the AHM mechanism and running 30 repeated experiments on Dataset IV. Figure 11 shows the deterioration in the best fitness compared to the full proposed model after removing the respective operator. Unsurprisingly, Task Migration is a crucial component for balancing task allocation, its removal resulted in an average fitness degradation of 0.61. Removing the other mutation operators led to a fitness degradation of 0.06 and 0.07. However, their contributions remain essential, as the diversity of strategies helps the model avoid local optima. This is further supported by the changing application probabilities of the mutation operators over the course of iteration, as illustrated in Figure 12. In the first 50 generations, when the population has not yet converged, Displacement and Task Migration can quickly break down structures, stimulate diversity, and actively optimize. In the middle phase (generations 50 to 200), as the model gradually stabilizes, Scramble introduces new structures to further expand the exploration of the solution space, while Inversion maintains a moderate level for fine-tuning. In the

final phase (after generation 200), the probabilities of Scramble and Task Migration decrease as the system favors stable structures, using the other operators for local refinement and expanding the solution boundaries.

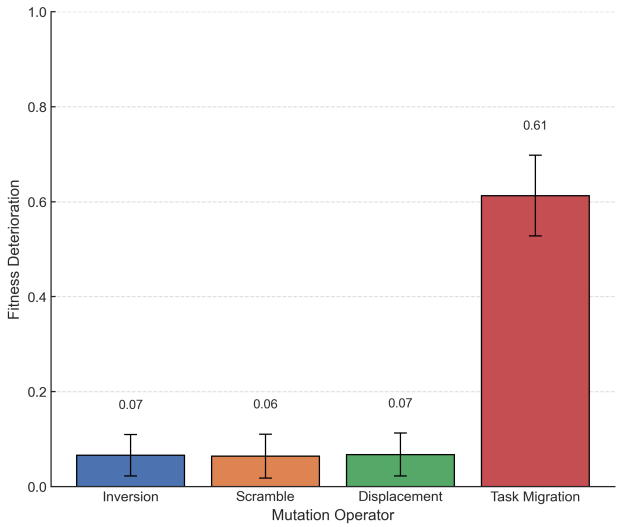


Figure 11. Deterioration of the best fitness after removing different mutation operators.

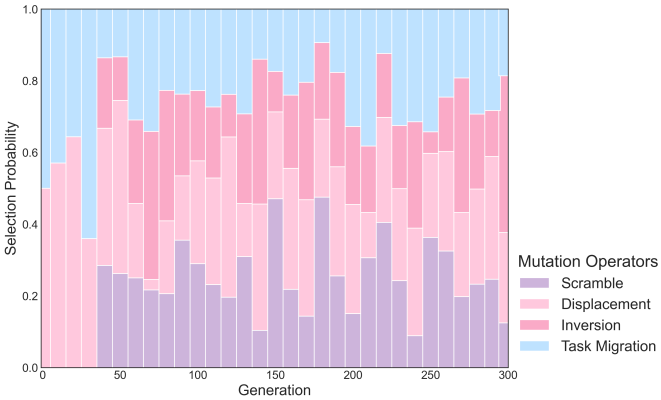


Figure 12. Change in the application probability of mutation operators during the iteration process.

5.2. Collaborative Trajectories

On 8 August 2023, a severe wildfire broke out on the island of Maui, Hawaii, causing extensive damage to buildings. Figure 13a shows the distribution of damaged buildings, mapped using machine learning (https://github.com/microsoft/building-damage-assessment, accessed on 1 June 2025.) combined with two periods of high-resolution satellite imagery. The survey areas were then delineated based on this map. Subsequently, the coverage path planning and the GA- Proposed multi-UAV scheduling and routing algorithm were used to plan the flight routes for a VM-UAV carrying three UAVs. The results are shown in Figure 13b. As seen in the figure, the scheduling and routing algorithm distributed the tasks relatively evenly among the UAVs, which undertook 20.9%, 35.4%, and 43.7% of the aerial survey tasks, respectively. Furthermore, the planned coverage paths within the grid respect the turning-radius constraints of fixed-wing UAVs, yielding more realistic and operationally feasible flight trajectories.

Zhoukou, Henan, is characterized by numerous, dispersed villages, which reflects the constraints and challenges that multi-UAV remote sensing applications may face in real-world settings. Aiming at village residential coverage, we evaluated the proposed genetic algorithm using six UAVs. The results are presented in Figure 14. As shown in Figure 14a,

the algorithm partitions the study area into four subregions. The northwest and southwest subregions contain fewer and more compact villages, so one UAV was assigned to each subregion and completed all visits without route intersections. The northeast and southeast subregions contain larger and more dispersed villages, so two UAVs were allocated to each. Figure 14b presents a Gantt chart of each UAV's flight plan, which indicates inter-region transfers and coverage trajectories. Transfers between regions consume relatively few resources, whereas most effort is devoted to coverage tasks, in accordance with the study objectives. Figure 14c shows the percentage task load for each UAV. The loads are relatively balanced and flight distances are similar, which supports the effectiveness of the proposed task-scheduling approach.

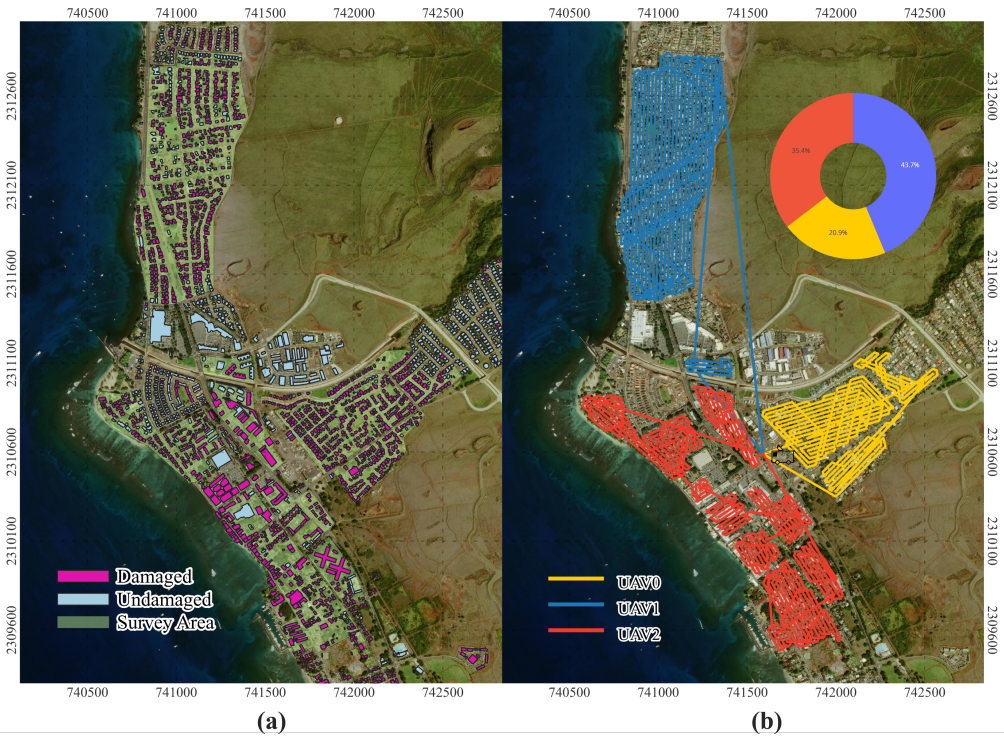


Figure 13. (**a**) Mapped distribution of damaged buildings and (**b**) results of the multi-UAV scheduling and coverage route planning in Maui, Hawaii.

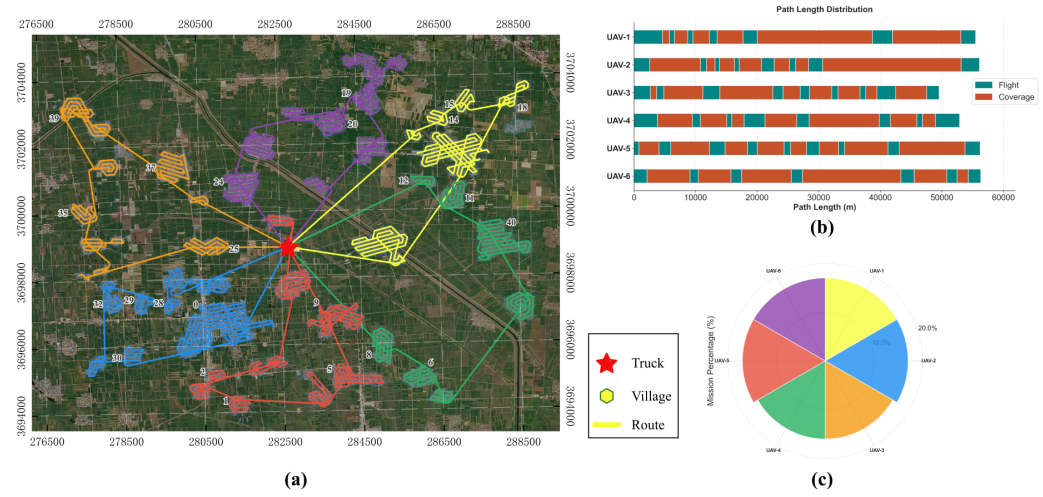


Figure 14. (**a**) UAV route and scheduling results for Henan, China. (**b**) Gantt chart of the mission paths. (**c**) Rose plot of the task load balance.

6. Discussion

To address the demands of collaborative UAV remote sensing, this study proposes a framework for cooperative VM-UAV task scheduling and coverage path planning. In practical applications, satellite imagery can be used to delineate the approximate survey regions. The proposed framework is then employed to generate optimized UAV scheduling schemes and flight trajectories. This approach facilitates the acquisition of high-resolution aerial orthophotos and 3D data while simultaneously minimizing the total flight distance and equalizing the workload distribution among the UAVs. Consequently, it significantly reduces the data acquisition window for large-scale surveys.

Despite its demonstrated advantages, the proposed framework primarily focuses on offline planning and task allocation, assuming an idealized static environment. Real-world UAV operations, however, necessitate the consideration of real-time physical information, dynamic communication topologies, and unexpected events such as obstacle appearance or rapidly deteriorating weather conditions. The current model does not account for these dynamic factors, which are critical for online mission execution. Furthermore, the energy consumption model is simplified and does not explicitly consider the complex impacts of factors like wind fields or other detailed aerodynamics. Future research will be directed at addressing these real-time operational requirements and extending the framework to support the practical application of heterogeneous UAV fleets.

7. Conclusions

This work addressed the complex, coupled problem of mission planning for Vehicle-mounted multi-UAV remote sensing systems. The core of our contribution is an advanced GA featuring two innovations: the MRECX and the AHM mechanism. MRECX proved effective in preserving and propagating high-quality path structures, accelerating convergence towards optimal solutions. The AHM mechanism dynamically balanced different mutation operators, enhancing the algorithm's ability to escape local optima and perform fine-grained searches, thereby improving overall performance and robustness. Comparative experiments demonstrated the superiority of our proposed GA over several benchmark algorithms, consistently achieving lower costs and better load balancing, especially in large-scale, complex scenarios. The practical utility of the methodology was validated through two diverse, real-world case studies, confirming its capability to generate efficient and practical flight plans for disaster assessment and agricultural mapping.

Although the proposed framework demonstrates clear advantages, it has several limitations. Future work should prioritize the incorporation of dynamic replanning capabilities to cope with environmental uncertainty. Extending the framework to support heterogeneous UAV fleets with varying capabilities (e.g., different sensor payloads and flight endurance) and integrating these platforms into a unified system—combined with multi-agent orthophoto generation and 3-D reconstruction for rapid mapping and perception—constitute important directions for further research.

Author Contributions: B.D.: conceptualization, methodology, writing—original draft, validation, software; A.T.: conceptualization, writing—review and editing; H.Y. (Huping Ye): formal analysis, investigation, project administration, writing—review and editing; H.Y. (Huanyin Yue): funding acquisition, writing—review and editing; C.X.: writing—review and editing; L.H.: writing—review and editing; H.H.: writing—review and editing; X.L.: conceptualization, funding acquisition, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (2023YFB3905705), the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDB0740100) and the Key Research and Development Program of Guangxi (GuikeAB25069501).

Data Availability Statement: Significance analysis records, vector maps delineating damaged buildings and the UAV survey area on Maui Island, Hawaii, as well as the vector data for the rural village survey area in Zhoukou (Henan Province), are available for download at https://drive.google.com/drive/folders/1hnDnL0JeVYDc4wxLLAd-9UetIky3mSYL?usp=sharing, accessed on 28 October 2025.

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

Appendix A

Table A1 summarizes the parameter settings for each algorithm. The notations used in the table are defined as follows: N_{pop} denotes the population size, $G_{\max}$ (or $I_{\max}$) represents the maximum number of iterations, P_c and P_m are the crossover and mutation probabilities, respectively. For PSO, S refers to the number of particles, ω, c_1, and c_2 indicate the inertia weight, cognitive acceleration coefficient, and social acceleration coefficient, respectively. For ACO, M refers to the number of ants, α and β are the weights of the pheromone and heuristic factors, and ρ denotes the pheromone evaporation rate. The selected values are based on parameter tuning to strike a balance between convergence speed and solution diversity.

Table A1. Simulation parameters for the algorithms.

Model	Parameters
GA-Default	$N_{pop} = 100$, $G_{\max} = 300$, $P_c = 0.8$, $P_m = 0.2$
GA-MRECX	$N_{pop} = 100$, $G_{\max} = 300$, $P_c = 0.7$, $P_m = 0.3$
GA-Proposed	$N_{pop} = 100$, $G_{\max} = 300$, $P_c = 0.5$, $P_m = 0.5$
PSO	$S = 100$, $I_{\max} = 300$, $\omega = 0.2$, $c_1 = 0.6$, $c_2 = 0.5$
ACO	$M = 100$, $I_{\max} = 300$, $\alpha = 1.0$, $\beta = 2.0$, $\rho = 0.1$
SC-TSA	$N_{pop} = 50$, $G_{max} = 150$, $P_c = 0.8$, $P_m = 0.2$

References

1. Ji, R.; Tan, K.; Wang, X.; Tang, S.; Sun, J.; Niu, C.; Pan, C. PatchOut: A novel patch-free approach based on a transformer-CNN hybrid framework for fine-grained land-cover classification on large-scale airborne hyperspectral images. *Int. J. Appl. Earth Obs. Geoinf.* **2025**, *138*, 104457. [CrossRef]
2. La Salandra, M.; Nicotri, S.; Donvito, G.; Italiano, A.; Colacicco, R.; Miniello, G.; Lapietra, I.; Roseto, R.; Dellino, P.; Capolongo, D. A paradigm shift in processing large UAV image datasets for emergency management of natural hazards. *Int. J. Appl. Earth Obs. Geoinf.* **2024**, *132*, 103996. [CrossRef]
3. Qin, K.; Li, J.; Zlatanova, S.; Wu, H.; Gao, Y.; Li, Y.; Shen, S.; Qu, X.; Yang, Z.; Zhang, Z.; et al. Novel UAV-based 3D reconstruction using dense LiDAR point cloud and imagery: A geometry-aware 3D gaussian splatting approach. *Int. J. Appl. Earth Obs. Geoinf.* **2025**, *140*, 104590. [CrossRef]
4. Guo, X.; Cheng, J.; Yin, C.; Li, Q.; Chen, R.; Fang, J. The extraordinary Zhengzhou flood of 7/20, 2021: How extreme weather and human response compounding to the disaster. *Cities* **2023**, *134*, 104168. [CrossRef]
5. Apostolidis, S.D.; Kapoutsis, P.C.; Kapoutsis, A.C.; Kosmatopoulos, E.B. Cooperative multi-UAV coverage mission planning platform for remote sensing applications. *Auton. Robot.* **2022**, *46*, 373–400. [CrossRef]
6. Liao, X.; Yue, H.; Liu, R.; Luo, X.; Luo, B.; Lu, M.; Ryan, B.; Ye, H. Launching an unmanned aerial vehicle remote sensing data carrier: Concept, key components and prospects. *Int. J. Digit. Earth* **2020**, *13*, 1172–1185. [CrossRef]
7. Skorobogatov, G.; Barrado, C.; Salamí, E. Multiple UAV systems: A survey. *Unmanned Syst.* **2020**, *8*, 149–169. [CrossRef]
8. Yakushiji, K.; Fujita, H.; Murata, M.; Hiroi, N.; Hamabe, Y.; Yakushiji, F. Short-range transportation using unmanned aerial vehicles (UAVs) during disasters in Japan. *Drones* **2020**, *4*, 68. [CrossRef]
9. Dong, Y.; Din, M.U.; Lagala, F.; Kuang, H.; Sun, J.; Yang, S.; Hussain, I.; He, S. Drone Carrier: An Integrated Unmanned Surface Vehicle for Autonomous Inspection and Intervention in GNSS-Denied Maritime Environment. *arXiv* **2025**, arXiv:2501.12869. [CrossRef]
10. Tahir, A.; Böling, J.; Haghbayan, M.H.; Toivonen, H.T.; Plosila, J. Swarms of unmanned aerial vehicles—A survey. *J. Ind. Inf. Integr.* **2019**, *16*, 100106. [CrossRef]

11. Javed, S.; Hassan, A.; Ahmad, R.; Ahmed, W.; Ahmed, R.; Saadat, A.; Guizani, M. State-of-the-art and future research challenges in uav swarms. *IEEE Internet Things J.* **2024**, *11*, 19023–19045. [CrossRef]
12. Liu, J.; Liao, X.; Ye, H.; Yue, H.; Wang, Y.; Tan, X.; Wang, D. UAV swarm scheduling method for remote sensing observations during emergency scenarios. *Remote Sens.* **2022**, *14*, 1406. [CrossRef]
13. Li, J.; Yang, X.; Yang, Y.; Liu, X. Cooperative mapping task assignment of heterogeneous multi-UAV using an improved genetic algorithm. *Knowl.-Based Syst.* **2024**, *296*, 111830. [CrossRef]
14. Muñoz, A.; Rubio, F. Evaluating genetic algorithms through the approximability hierarchy. *J. Comput. Sci.* **2021**, *53*, 101388. [CrossRef]
15. Pasha, J.; Elmi, Z.; Purkayastha, S.; Fathollahi-Fard, A.M.; Ge, Y.E.; Lau, Y.Y.; Dulebenets, M.A. The drone scheduling problem: A systematic state-of-the-art review. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 14224–14247. [CrossRef]
16. Hu, M.; Liu, W.; Peng, K.; Ma, X.; Cheng, W.; Liu, J.; Li, B. Joint routing and scheduling for vehicle-assisted multidrone surveillance. *IEEE Internet Things J.* **2018**, *6*, 1781–1790. [CrossRef]
17. Kim, S.; Moon, I. Traveling salesman problem with a drone station. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 42–52. [CrossRef]
18. Kitjacharoenchai, P.; Ventresca, M.; Moshref-Javadi, M.; Lee, S.; Tanchoco, J.M.; Brunese, P.A. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Comput. Ind. Eng.* **2019**, *129*, 14–30. [CrossRef]
19. Chen, J.; Xie, J. Joint task scheduling, routing, and charging for multi-UAV based mobile edge computing. In Proceedings of the ICC 2022-IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022 ; IEEE: Piscataway, NJ, USA , 2022; pp. 1–6.
20. Hong, D.; Lee, S.; Cho, Y.H.; Baek, D.; Kim, J.; Chang, N. Energy-Efficient Online Path Planning of Multiple Drones Using Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9725–9740. [CrossRef]
21. Mao, X.; Wu, G.; Fan, M.; Cao, Z.; Pedrycz, W. DL-DRL: A Double-Level Deep Reinforcement Learning Approach for Large-Scale Task Scheduling of Multi-UAV. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 1028–1044. [CrossRef]
22. Panowicz, R.; Stecz, W. Robust Optimization Models for Planning Drone Swarm Missions. *Drones* **2024**, *8*, 572. [CrossRef]
23. Xing, X.; Zhou, Z.; Li, Y.; Xiao, B.; Xun, Y. Multi-UAV Adaptive Cooperative Formation Trajectory Planning Based on an Improved MATD3 Algorithm of Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 12484–12499. [CrossRef]
24. Shima, T.; Rasmussen, S.J.; Sparks, A.G. UAV cooperative multiple task assignments using genetic algorithms. In Proceedings of the 2005, American Control Conference, Portland, OR, USA, 8–10 June 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 2989–2994.
25. Zhu, W.; Li, L.; Teng, L.; Yonglu, W. Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding. *Chin. J. Aeronaut.* **2018**, *31*, 339–350.
26. Yan, F.; Chu, J.; Hu, J.; Zhu, X. Cooperative task allocation with simultaneous arrival and resource constraint for multi-UAV using a genetic algorithm. *Expert Syst. Appl.* **2024**, *245*, 123023. [CrossRef]
27. Gao, Y.; Zhang, Y.; Zhu, S.; Sun, Y. Multi-UAV task allocation based on improved algorithm of multi-objective particle swarm optimization. In Proceedings of the 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Zhengzhou, China, 18–20 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 443–4437.
28. Li, Y.; Zhang, Z.; Sun, Q.; Huang, Y. An improved ant colony algorithm for multiple unmanned aerial vehicles route planning. *J. Frankl. Inst.* **2024**, *361*, 107060. [CrossRef]
29. Choset, H.; Pignon, P. Coverage path planning: The boustrophedon cellular decomposition. In *Field and Service Robotics*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 203–209.
30. Torres, M.; Pelta, D.A.; Verdegay, J.L.; Torres, J.C. Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. *Expert Syst. Appl.* **2016**, *55*, 441–451. [CrossRef]
31. Shah, K.; Ballard, G.; Schmidt, A.; Schwager, M. Multidrone aerial surveys of penguin colonies in Antarctica. *Sci. Robot.* **2020**, *5*, eabc3000. [CrossRef] [PubMed]
32. Kapoutsis, A.C.; Chatzichristofis, S.A.; Kosmatopoulos, E.B. DARP: Divide areas algorithm for optimal multi-robot coverage path planning. *J. Intell. Robot. Syst.* **2017**, *86*, 663–680. [CrossRef]
33. Cao, Y.; Cheng, X.; Mu, J. Concentrated coverage path planning algorithm of UAV formation for aerial photography. *IEEE Sensors J.* **2022**, *22*, 11098–11111. [CrossRef]
34. Krupke, D. Near-optimal coverage path planning with turn costs. In Proceedings of the 2024 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX), SIAM, Alexandria, VA, USA, 7–8 January 2024; pp. 118–132.
35. Kolmogorov, V. Blossom V: A new implementation of a minimum cost perfect matching algorithm. *Math. Program. Comput.* **2009**, *1*, 43–67. [CrossRef]
36. Whitley, L.D.; Starkweather, T.; Fuquay, D. Scheduling problems and traveling salesmen: The genetic edge recombination operator. In Proceedings of the 3rd International Conference on Genetic Algorithms, Fairfax, VA, USA, 2–9 June 1989; pp. 133–140.
37. Goldberg, D.E.; Lingle, R. Alleles, loci, and the traveling salesman problem. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*; Psychology Press: London, UK, 2014; pp. 154–159.

38. Zhang, C.; Jiang, Z.; Tang, H.; Song, G. Path Planning for Multi-UAV Target Coverage via Improved GA-PSO Algorithm. In Proceedings of the 2024 6th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI), Nanjing, China, 6–8 December 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 304–310.
39. Shafiq, M.; Ali, Z.A.; Israr, A.; Alkhammash, E.H.; Hadjouni, M.; Jussila, J.J. Convergence analysis of path planning of multi-UAVs using max-min ant colony optimization approach. *Sensors* **2022**, *22*, 5395. [CrossRef] [PubMed]
40. Kruskal, W.H.; Wallis, W.A. Use of Ranks in One-Criterion Variance Analysis. *J. Am. Stat. Assoc.* **1952**, *47*, 583–621. [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.