electronics

MDPI

Article

TeCoR-UAV: A Two-Stage Topology Extraction and Cooperative Routing Algorithm for Low-Altitude Logistics

Buyang Ding [1], Weijun Ni [1], Yixing Luo [2], Zhiming Liu [1], Nianyu Li [3], Jialong Li [4] and Mingyue Zhang [1,*]

1 College of Computer and Information Science, Southwest University, Chongqing 400715, China
2 Software Department, Beijing Institute of Control Engineering, Beijing 100190, China
3 ZGC Laboratory, Beijing 100190, China
4 Waseda Institute for Advanced Study, Waseda University, Tokyo 169-0051, Japan
* Correspondence: myzhangswu@swu.edu.cn

Abstract

Multi-UAV cooperative delivery is a key technology for intelligent low-altitude logistics, with applications in mountainous-area transport, urban last-mile delivery, and emergency resupply. In complex three-dimensional (3D) low-altitude environments, obstacle-constrained airspace, fleet heterogeneity, payload limits, and time windows make the realistic representation of flight costs difficult and substantially restrict the feasible region of cooperative planning. To address these challenges, this paper proposes TeCoR-UAV, a two-stage topology extraction and cooperative route planning framework. The proposed method first precomputes executable flight trajectories in obstacle-constrained airspace and constructs a topological graph that captures realistic flight costs. A bi-objective optimization model is then formulated to minimize operational cost and maximize service quality. Furthermore, a hierarchical genetic solver is designed to improve solution quality and feasibility jointly through global task allocation and single-UAV execution sequence optimization. Experimental results show that the proposed method can better reflect realistic flight costs in complex environments. Compared with existing benchmark methods, TeCoR-UAV achieves better bi-objective trade-offs in most medium- and large-scale scenarios, as well as in topologically constrained scenarios, and improves service quality by an average of 18.5 percentage points, indicating its scenario adaptability and potential for practical application.

Keywords: multi-UAV systems; intelligent route planning; cooperative optimization; low-altitude logistics

Academic Editor: Mahmut Reyhanoglu

Received: 1 May 2026
Revised: 24 June 2026
Accepted: 30 June 2026
Published: 5 July 2026

1. Introduction

Low-altitude logistics has emerged as an important development direction in intelligent transportation and smart logistics, and UAV-based freight delivery is one of its most representative application forms [1,2]. Recent advances in autonomous navigation, wireless communication, and artificial intelligence have accelerated the deployment of UAVs in low-altitude logistics, establishing them as an important platform for intelligent delivery services [3]. Compared to conventional ground transportation, UAVs offer greater spatial flexibility and can perform short- to medium-range, time-sensitive, and small-batch delivery tasks more efficiently. As they are not directly constrained by the existing road network, they exhibit clear advantages in remote areas, complex terrain, and other scenarios with limited ground accessibility [4]. Furthermore, UAV-based delivery also shows strong potential to reduce labor dependence, shorten delivery time, and improve operational efficiency, while its low energy consumption and reduced emissions are consistent with the broader trend towards green and low-carbon logistics [3,5]. These advantages have motivated growing interest in low-altitude logistics applications, including medical emergency transport, last-mile e-commerce delivery, and post-disaster supply distribution [6–8].

Despite the significant advantages of UAVs in low-altitude logistics, mission execution remains jointly constrained by complex operating environments and multidimensional operational requirements. In the spatial domain, continuously undulating mountain terrain and densely distributed urban buildings form complex unstructured obstacle fields, which substantially increase the difficulty of obstacle avoidance and route planning [9]. In the operational domain, delivery tasks must simultaneously account for heterogeneous UAV capabilities, payload-capacity constraints, load-coupled flight performance, and service time-window considerations [10]. The coupling of these factors makes realistic flight costs difficult to accurately represent and significantly reduces the feasible region of cooperative planning. As a result, low-altitude logistics is no longer a simple shortest-path search or task-scheduling problem, but rather a highly constrained decision-making problem that must balance safety, timeliness, and economic efficiency. In complex 3D low-altitude environments, how to faithfully represent flight accessibility and realistic flight cost, and on this basis generate executable cooperative delivery plans for heterogeneous UAV fleets, has become a key issue that urgently needs to be addressed.

Existing studies on multi-UAV cooperative delivery can be broadly divided into two categories. The first focuses on task allocation and cooperative scheduling, typically by formulating variants of the vehicle routing problem (VRP) and solving them with heuristic or hybrid optimization methods to improve system-level efficiency and resource utilisation [11]. The second emphasizes path planning in complex flight environments, with particular attention to obstacle avoidance, trajectory feasibility, and flight safety [12,13]. However, the connection between these two research streams remains limited. On the one hand, many cooperative scheduling approaches still estimate inter-node flight costs under two-dimensional assumptions or idealized airspace abstractions, and therefore fail to capture the actual costs induced by complex 3D terrain and obstacle constraints. On the other hand, although many 3D path-planning methods can generate locally feasible trajectories, they do not explicitly incorporate logistics-related constraints such as fleet heterogeneity, payload–energy coupling, delivery time windows, and multi-UAV coordination. As a consequence, current studies still lack a unified framework that integrates realistic 3D flight-cost representation with cooperative delivery decision-making. However, such integration is intrinsically difficult, as path generation, task assignment, and cooperative conflict avoidance become tightly coupled; when these decisions are solved jointly in a continuous 3D space, the search space grows rapidly and the computational burden increases substantially [14]. This gap continues to limit the practical applicability of existing methods in complex low-altitude logistics scenarios.

To address this issue, this paper proposes TeCoR-UAV, a two-stage framework for topological extraction and cooperative route planning. In the first stage, traversable free space is explored under kinematic and safety constraints to construct a collision-free skeletal network. Through spatial mapping and topological compression, this network is then transformed into a topological graph, whose nodes, edges, and edge weights provide a compact surrogate representation of the original airspace. In the second stage, the resulting topological graph is used to formulate a bi-objective optimization model that minimizes operational cost while maximizing service quality, subject to payload limits, UAV range constraints, energy-consumption fluctuations, delivery time windows, and fleet coordination constraints. To solve the resulting strongly constrained combinatorial problem, a hierarchical genetic solver is developed to jointly perform task allocation and route planning on the topological graph, thereby producing cooperative delivery strategies that are both efficient and operationally feasible.

The main contributions of this study are as follows.

(1) We propose an environment-aware topological routing interface in TeCoR-UAV to connect executable 3D flight trajectories with high-level multi-UAV cooperative routing decisions. Unlike existing approaches that rely on idealized Euclidean-distance approximations or treat 3D obstacle avoidance as an isolated post-processing module, TeCoR-UAV transforms executable 3D obstacle avoidance trajectories into a reusable weighted topological representation and embeds environmental awareness into the cooperative routing process. As a result, realistic 3D flight costs can directly participate in task allocation and routing decisions, reducing repeated continuous-space path searches while maintaining consistency between physical feasibility and cooperative decision-making.

(2) We propose a structure-preserving, weakly coupled hierarchical co-evolution mechanism to address the strong coupling and sparse feasible region in multi-UAV cooperative delivery. Unlike conventional fully decoupled procedures, TeCoR-UAV jointly represents task assignment and coarse-grained visiting order in the global search layer, enabling lower-level solutions to preserve a feasible coarse routing structure. The upper layer then refines the single-UAV execution sequences under fixed assignment structures, thereby achieving an effective division of labor between cross-UAV allocation and intra-UAV route optimization. Through cross-layer information transfer and joint solution-set filtering, the proposed architecture reduces the practical difficulty of joint search and alleviates the assignment-route inconsistency problem.

(3) We formulate a multi-UAV cooperative logistics delivery model and establish a simulation-based validation platform for complex low-altitude logistics scenarios. Systematic evaluations are conducted in seven test scenarios that cover mountainous, urban, and topologically constrained environments, with problem scales of up to 50 task nodes. TeCoR-UAV is compared with six representative benchmark algorithms. The results show that TeCoR-UAV achieves favorable bi-objective trade-off performance in most medium- and large-scale scenarios, as well as in topology-constrained scenarios, thereby validating its effectiveness and scenario adaptability.

The rest of this paper is structured as follows: We first review related work to clarify the problem background and identify existing research gaps. Next, the mathematical formulation of the multi-UAV cooperative delivery problem is presented. Building on this, we detail the proposed two-stage TeCoR-UAV framework. Subsequently, extensive simulation experiments are presented to validate the effectiveness of the proposed approach. Finally, we conclude the paper and outline directions for future research.

2. Related Work

The effectiveness of UAV logistics delivery depends to a large extent on coordinated optimization of task allocation and route planning, as these decisions directly affect delivery efficiency, operational cost, and service reliability [15,16]. Existing studies can be broadly categorized into two research streams: UAV path planning in complex 3D environments and strongly constrained cooperative optimization for multi-UAV logistics delivery.

2.1. UAV Path Planning in Complex 3D Environments

Research on UAV path planning in complex 3D environments is primarily concerned with the generation of safe and feasible flight trajectories between prescribed origin–destination pairs. Existing methods can generally be grouped into three categories: graph-search-based methods, sampling-based planners, and intelligent optimization approaches.

Graph-search-based methods have been widely used to efficiently generate feasible paths in discretized spaces, with A*-based approaches remaining among the most representative solutions [12,17]. In addition, intelligent optimization methods have been introduced, including artificial bee colony, gray wolf optimizer, genetic algorithm, eagle search, and ant colony optimization, to improve global search performance and path feasibility in complex environments. These methods typically incorporate hybrid evolutionary mechanisms, waypoint encoding schemes, and space partitioning strategies to improve solution quality [18–22]. Sampling-based planners, particularly those derived from rapidly exploring random trees (RRT), have also been extensively applied to UAV path planning [13,23,24]. By integrating mechanisms such as potential-guided sampling, lazy collision checking, and closed-loop control, these methods have improved search efficiency, feasibility, and robustness in complex 3D scenarios.

More recently, research has progressed beyond the single-path search toward 3D airspace discretization, airway-network construction, and topological environment representation, to strengthen the capability of environmental modeling and improve planning efficiency in dense urban settings [25]. Nevertheless, most of these studies remain focused on the avoidance of single-UAV obstacles or the generation of a local trajectory. Although they are effective in handling terrain variations, building obstacles, and flight-smoothness constraints, they do not explicitly support the cooperative decision-making required in multi-UAV logistics delivery. Therefore, their contribution to cooperative route planning at the system-level remains limited.

2.2. Strongly Constrained Cooperative Optimization for Multi-UAV Logistics Delivery

Compared with single-UAV path planning, multi-UAV logistics delivery requires joint consideration of task allocation, route planning, temporal coordination, and resource management, and is therefore inherently a strongly constrained joint optimization problem. Most existing studies formulate this problem using the vehicle routing problem (VRP) or its variants, and solve it through evolutionary algorithms, mixed-integer programming, reinforcement learning, or heuristic strategies. These approaches have achieved promising results in terms of mission completion time, operational cost, and resource utilization [26–34].

As application scenarios have become more complex, the constraints considered in the literature have gradually expanded from simple formulations oriented towards distance or time to richer operational settings involving heterogeneous UAV fleets, delivery time windows, energy-consumption heterogeneity, endurance limits, repeated returns to depots, and risk-control requirements [35–37]. In addition, some recent studies have begun to examine the coupling between task allocation and route planning, and have adopted staged solution mechanisms to improve computational efficiency and coordination quality in complex scenarios [38–40].

Although substantial progress has been made, two key challenges remain in complex low-altitude logistics scenarios. First, most existing studies still rely on the two-dimensional Euclidean distance or simplified graph abstractions to approximate the cost of node-to-node travel. This makes it difficult to capture the detour cost and accessibility constraints induced by 3D terrain and obstacle fields, and may therefore produce cooperative solutions with limited executability in real airspace. Second, logistics delivery is governed by tightly coupled operational constraints, including heterogeneous resources, payload–energy coupling, and delivery time windows, all of which substantially reduce the feasible solution space. For such problems with sparse feasible regions, exact mathematical programming methods often incur prohibitive computational costs in large-scale settings. Conventional evolutionary and heuristic methods, meanwhile, generally lack effective mechanisms for feasibility preservation and repair, whereas learning-based methods still face limitations in hard-constraint satisfaction and cross-scenario generalization.

These limitations indicate the need for a unified framework that can represent both realistic airspace accessibility and flight costs in a compact form and support efficient cooperative optimization in sparse feasible regions.

3. Problem Formulation

3.1. Task Scenario Description

Figure 1 illustrates a representative multi-UAV delivery-and-return scenario. The system consists of a central depot, three terminal logistics service points, two heterogeneous UAVs, and multiple obstacles distributed within the airspace. Each UAV is equipped with two independent cargo compartments: one for carrying outbound delivery items required by the service points, and the other for storing returned goods collected during mission execution. In practical low-altitude logistics operations, each service node is regarded as a terminal logistics agent that supports parcel handover and can provide battery replacement for UAVs when required.

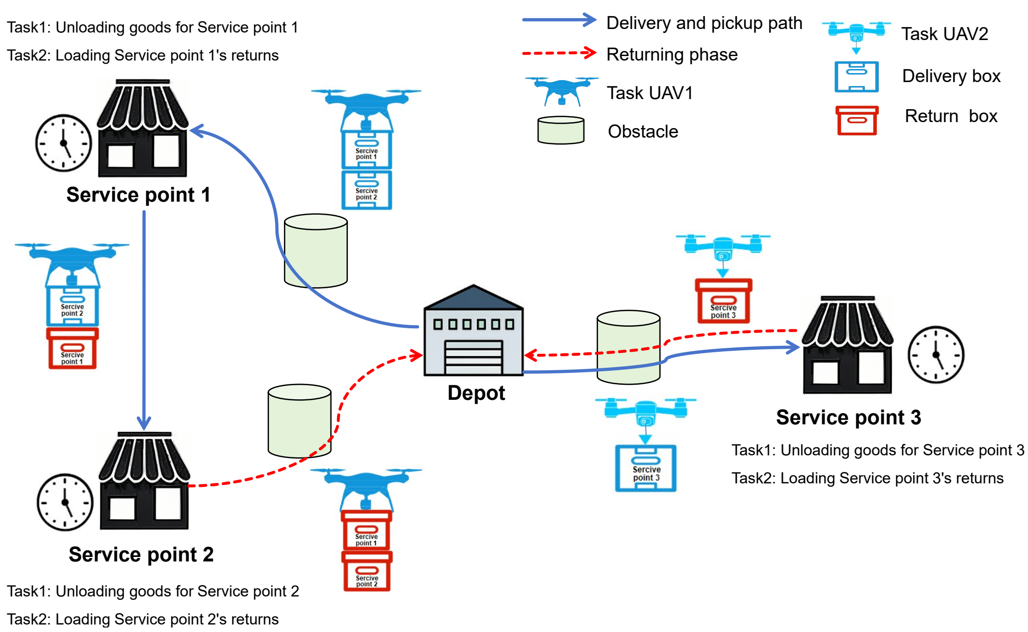

Figure 1. Representative multi-UAV delivery-and-pickup scenario in obstacle-constrained airspace.

The depot serves as both the origin and terminal node of the mission. Before departure, the UAVs are loaded with the requested delivery items at the depot; upon completion of the mission, the depot receives the returned goods collected from the service points. The three service points are geographically distributed, and each is associated with a coupled service request that includes item delivery and return collection, together with a prescribed delivery time window. The obstacles shown in the figure represent no-fly regions in the airspace. The connecting lines denote the UAV flight routes during task execution, where solid lines indicate routes used for delivery and return services, and dashed lines denote return-to-depot routes.

Before mission execution, the depot generates a task allocation plan for the UAV fleet according to logistics service requests. Each UAV then visits the subset of service points assigned to it. At each service point, the UAV first unloads the requested delivery items and subsequently collects the returned items specified by that service point. After completion of all assigned services, the UAV follows the planned route back to the depot.

3.2. Physical Environment Modelling

In UAV logistics delivery, the modeling of the physical environment provides the basis for both route planning and task scheduling. This study places particular emphasis on two representative classes of operating environments: mountainous and urban environments.

3.2.1. Mountainous Environment Model

The mountainous flight environment is represented by a mathematical model, as defined in Equation (1).

$$Z(x,y) = \sum_{i=1}^{n_{peak}} H_i \exp\left(-\frac{(x-x_i)^2}{w_{x_i}^2} - \frac{(y-y_i)^2}{w_{y_i}^2} \right) \tag{1}$$

where (x_i, y_i) denotes the center coordinates of the i-th mountain, H_i is its peak height, and w_{x_i} and w_{y_i} characterize its spatial spread along the x- and y-directions, respectively. Here, n_{peak} denotes the total number of mountains. The resulting mountain environment model is illustrated in Figure 2a.

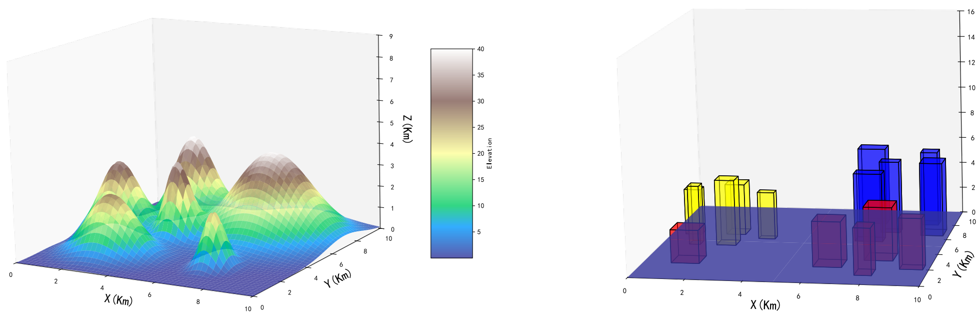

Figure 2. Environment models: (**a**) Mountain environment. (**b**) Urban environment.

3.2.2. Urban Environment Model

In this study, an environment model based on 3D cuboids is constructed to represent urban buildings. Each building is defined by a six-tuple, $(x_{\text{build}}, y_{\text{build}}, z_{\text{build}}, L, W, H)$, where $(x_{\text{build}}, y_{\text{build}}, z_{\text{build}})$ denotes the coordinates of the lower-left corner of the building, L denotes the building length along the x-axis, W denotes the building length along the y-axis, and H denotes the building height.

The urban environment model is composed of a set of discrete buildings and is denoted by

$$\mathcal{O} = \left\{ \left(x_{\text{build}}^{(k)}, y_{\text{build}}^{(k)}, z_{\text{build}}^{(k)}, L^{(k)}, W^{(k)}, H^{(k)} \right) \in \mathbb{R}^6 \,\middle|\, k = 1, 2, \ldots, N_{\text{obs}} \right\},$$

where N_{obs} denotes the total number of buildings. The resulting urban environment model is illustrated in Figure 2b.

3.3. Modelling of the Multi-UAV Delivery Problem

3.3.1. Notation and Decision Variables

For a rigorous formulation of the model, we first define the sets, parameters, and decision variables involved in the multi-UAV delivery-and-collection problem. The principal notation adopted throughout this study is summarized in Table 1.

Table 1. Sets, parameters, and decision variables in the multi-UAV delivery-and-pickup model.

Variable	Description
C	Set of service points, $C = \{1, 2, ..., n\}$
K	Set of UAVs, $K = \{1, 2, ..., m\}$
V	Set of nodes, $V = \{0\} \cup C \cup \{n+1\}$, where O_0 and O_{n+1} denote the departure depot and the arrival depot, respectively.
A	Set of feasible arcs on the topological graph
d_{ij}	Shortest flight distance from node i to node j
q_i	Delivery demand of service point i
r_i	Pickup demand of service point i
h_i	Service time at service point i
$[a_i^-, a_i, b_i, b_i^+]$	Fuzzy trapezoidal time window of service point i
Q_k	Maximum payload capacity of UAV k
v_k^0	Unloaded cruising speed of UAV k
e_k^0	Unit energy-consumption parameter of UAV k under no-load conditions
η_{ij}^k	Energy consumption rate per unit distance of UAV k on arc (i, j)
$R_k^{\max}$	Maximum full-battery flight range of UAV k
f_k	Fixed activation cost of UAV k
ma_k	The maintenance cost of UAV k
ρ	Unit energy-cost coefficient
α	Payload-dependent speed attenuation coefficient
β	Payload-dependent energy-consumption amplification coefficient
τ	Unit handling time per kg
τ^{swap}	Fixed battery-swapping time at a service point
M	A sufficiently large positive constant
$x_{ij}^k \in \{0, 1\}$	Equals 1 if UAV k flies directly from node i to node j, and 0 otherwise
$y_k \in \{0, 1\}$	Equals 1 if UAV k is activated, and 0 otherwise
$y_i^k \in \{0, 1\}$	Equals 1 if UAV k performs battery swapping at service point i before leaving the node, and 0 otherwise
$z_i^k \in \{0, 1\}$	Equals 1 if service point i is served by UAV k, and 0 otherwise
$u_i^k \geq 0$	Sequencing variable of service point i in the route of UAV k
$A_i \geq 0$	Arrival time at service point i
$T_i \geq 0$	Time at which unloading at service point i is completed
$B_i \geq 0$	Departure time from service point i after service completion and possible battery swapping
$B_0^k \geq 0$	Departure time of UAV k from the depot after initial loading
$A_{n+1}^k \geq 0$	Arrival time of UAV k at the terminal depot
$W_i^k \geq 0$	Real-time payload of UAV k upon leaving node i
$E_{ij}^k \geq 0$	Energy consumption of UAV k on arc (i, j)
$s_i \in [0, 1]$	Satisfaction level of service point i

3.3.2. Load-Coupled Models of UAV Energy Consumption, Flight Speed, and Temporal Propagation

In low-altitude logistics operations, the flight speed of the UAV, energy consumption, and temporal propagation are affected by the instantaneous payload. Consequently, we formulate load-coupled models to capture these payload-dependent effects.

(1) **Payload–speed model.** Given that the cruising speed of a rotary-wing UAV decreases as its instantaneous payload increases, the actual cruising speed of the UAV k in

the arc (i, j), denoted by v_{ij}^k, is modeled as a linearly decreasing function of the instantaneous payload W_i^k upon departure from node i, as shown in Equation (2).

(2) **Payload–energy consumption model and energy cost.** The flight energy consumption of a UAV consists of two components: baseline energy expenditure and the additional energy induced by the payload. Let η_{ij}^k denote the rate of energy consumption per unit of distance of the UAV k on the arc (i, j). This quantity is formulated as a linear function of the instantaneous payload W_i^k, as given in Equation (3). Consequently, the total energy consumption of the flight in the arc (i, j), denoted by E_{ij}^k, can be expressed as the product of the flight distance and the rate of energy consumption per unit of distance, as shown in Equation (4).

(3) **Temporal propagation model.** The mission duration of a UAV comprises depot loading, in-flight travel, unloading, pickup loading, node-level service dwell time, and possible battery replacement time. For UAV k, the departure time from the depot is determined by its initial loading quantity, as defined in Equation (5). Thereafter, the arrival time at each subsequent node is propagated recursively from the departure time of the preceding node and the corresponding flight time, as described in Equations (6)–(8). For the service point i, the completion time of the delivery task is defined as the instant at which the unloading is complete, as given in Equation (9). The departure time after service is then jointly determined by the unloading process, the pickup loading, the prescribed dwell time, and the battery replacement time if replacement is performed at the node, as expressed in Equation (10).

It should be noted that the linear payload–speed and payload–energy formulations are simplified first-order approximations for routing optimization. Although multirotor UAV energy consumption may depend nonlinearly on take-off weight, flight speed, rotor characteristics, and operating conditions, the linear payload-dependent model is adopted here to preserve the tractability and interpretability of the collaborative routing formulation. Similar approximations have also been used in drone delivery routing studies within limited payload ranges [15,41].

$$v_{ij}^k = v_k^0 - \alpha W_i^k, \quad v_{ij}^k > 0, \quad \forall (i, j) \in A, \ \forall k \in K \tag{2}$$

$$\eta_{ij}^k = e_k^0 + \beta W_i^k, \quad \forall (i, j) \in A, \ \forall k \in K \tag{3}$$

$$E_{ij}^k = d_{ij}\left(e_k^0 + \beta W_i^k\right) x_{ij}^k, \quad \forall (i, j) \in A, \ \forall k \in K \tag{4}$$

$$B_0^k = \tau \sum_{i \in C} q_i z_i^k, \quad \forall k \in K \tag{5}$$

$$A_j \geq B_0^k + \frac{d_{0j}}{v_k^0 - \alpha W_0^k} - M(1 - x_{0j}^k), \quad \forall (0, j) \in A, \ j \in C, \ \forall k \in K \tag{6}$$

$$A_j \geq B_i + \frac{d_{ij}}{v_k^0 - \alpha W_i^k} - M(1 - x_{ij}^k), \quad \forall (i, j) \in A, \ i, j \in C, \ i \neq j, \ \forall k \in K \tag{7}$$

$$A_{n+1}^k \geq B_i + \frac{d_{i,n+1}}{v_k^0 - \alpha W_i^k} - M(1 - x_{i,n+1}^k), \quad \forall (i, n+1) \in A, \ i \in C, \ \forall k \in K \tag{8}$$

$$T_i = A_i + \tau q_i, \quad \forall i \in C \tag{9}$$

$$B_i = A_i + \tau q_i + \tau r_i + h_i + \tau^{swap} \sum_{k \in K} y_i^k, \quad \forall i \in C \tag{10}$$

3.3.3. Scope of the Optimization Model

The optimization model is formulated on a fixed weighted topological graph within a single planning horizon. Terrain, buildings, static obstacles, known no-fly regions, graph connectivity, and edge weights are treated as given inputs before routing optimization.

Thus, the model focuses on global cooperative task allocation and route sequencing under static or quasi-static low-altitude airspace conditions.

Known changes in airspace restrictions, such as temporary no-fly zones announced before take-off, are incorporated by updating the input topological graph before a new planning round. The routing model does not explicitly treat newly imposed no-fly zones during flight execution as time-dependent decision constraints.

Dynamic obstacles, such as other aircraft, are handled in the local execution layer rather than in the global routing model. Each topological edge represents a precomputed feasible inter-node trajectory under known environmental conditions, while short-range deconfliction during flight is assumed to be managed by on-board collision-avoidance or mutual-exclusion safety modules. Thus, the proposed model provides global route-level coordination, whereas local flight safety is ensured by UAV platform-level safety-control functions in practical deployment.

3.4. Objective Functions and Constraints

The multi-UAV collaborative delivery problem is formulated as a bi-objective optimization model that minimizes total operating cost and maximizes overall service quality.

(1) Objective functions

(a) Minimization of total operating cost. The objective F_1 aims to minimize the total operating cost, including the flight energy cost, the UAV activation cost, and the maintenance cost, as defined in Equation (11).

(b) Maximization of service quality. The objective F_2 aims to maximize overall service quality, where service-point satisfaction is described by the fuzzy trapezoidal time-window function in Equation (12), and the aggregate satisfaction objective is given in Equation (13).

(2) Constraints

To ensure the operational feasibility of the collaborative multi-UAV delivery scheme, the model considers service assignment, route structure, temporal propagation, payload capacity, UAV range feasibility, battery-swapping feasibility, and variable-domain constraints, as formulated in Equations (14)–(30).

Equation (14) ensures that each service point is served exactly once by a single UAV. Equations (15) and (16) impose node-flow balance constraints, requiring the assigned UAV to enter and leave the corresponding service point exactly once. Equations (17) and (18) define the departure–return structure, ensuring that each activated UAV departs from the origin depot and returns to the destination depot. Equations (19) and (20) eliminate subtours and guarantee complete depot-connected delivery routes. Equations (21) and (22) specify the initial loading conditions, where the initial payload of each UAV is determined by the assigned delivery demand and must not exceed its payload capacity. Equations (23)–(25) describe the evolution of the payload and the capacity constraints along each route, accounting for both unloading the delivery items and loading the goods returned. Equations (26)–(30) describe UAV range-feasibility and battery swap logic during route evaluation. Each activated UAV starts with a full battery range $R_k^{\max}$, and each traveled arc must not exceed this maximum range. If the remaining range before leaving the service point i is sufficient to reach the node j, the UAV flies directly to j, and the remaining range is reduced by d_{ij}. Otherwise, battery swapping is performed before departure, the remaining range is reset to $R_k^{\max}$, and the fixed swapping time is included in the temporal propagation model through $y_i^k \tau^{swap}$. If $d_{ij} > R_k^{\max}$, the arc is infeasible for UAV k because it cannot be reached even with a fully charged battery. Consequently, in the formal model developed in this study, service time windows are handled through the satisfaction objective in Equations (12) and (13), whereas feasibility is enforced mainly through service-point assignment, route structure, temporal propagation, payload capacity, and UAV range feasibility constraints in Equations (14)–(30).

$$\min F_1 = \rho \sum_{k\in K} \sum_{(i,j)\in A} E_{ij}^k + \sum_{k\in K} (f_k + ma_k) y_k \tag{11}$$

$$s_i = \begin{cases} \dfrac{T_i - a_i^-}{a_i - a_i^-}, & a_i^- \le T_i < a_i, \\ 1, & a_i \le T_i \le b_i, \\ \dfrac{b_i^+ - T_i}{b_i^+ - b_i}, & b_i < T_i \le b_i^+, \\ 0, & \text{otherwise}, \end{cases} \quad \forall i \in C \tag{12}$$

$$\max F_2 = \frac{1}{n} \sum_{i\in C} s_i \tag{13}$$

$$\sum_{k\in K} z_i^k = 1, \qquad \forall i \in C \tag{14}$$

$$\sum_{\substack{j\in C\cup\{n+1\} \\ (i,j)\in A}} x_{ij}^k = z_i^k, \quad \forall i \in C, \forall k \in K \tag{15}$$

$$\sum_{\substack{j\in\{0\}\cup C \\ (j,i)\in A}} x_{ji}^k = z_i^k, \quad \forall i \in C, \forall k \in K \tag{16}$$

$$\sum_{\substack{j\in C \\ (0,j)\in A}} x_{0j}^k = y_k, \quad \forall k \in K \tag{17}$$

$$\sum_{\substack{i\in C \\ (i,n+1)\in A}} x_{i,n+1}^k = y_k, \quad \forall k \in K \tag{18}$$

$$u_j^k \ge u_i^k + 1 - (n+1)(1 - x_{ij}^k), \quad \forall (i,j) \in A,\ i,j \in C,\ i \ne j,\ \forall k \in K \tag{19}$$

$$z_i^k \le u_i^k \le n z_i^k, \quad \forall i \in C, \forall k \in K \tag{20}$$

$$W_0^k = \sum_{i\in C} q_i z_i^k, \qquad \forall k \in K \tag{21}$$

$$0 \le W_0^k \le Q_k y_k, \qquad \forall k \in K \tag{22}$$

$$W_j^k \ge W_i^k - q_j + r_j - M(1 - x_{ij}^k), \quad \forall (i,j) \in A,\ i \in \{0\} \cup C,\ j \in C,\ \forall k \in K \tag{23}$$

$$W_j^k \le W_i^k - q_j + r_j + M(1 - x_{ij}^k), \quad \forall (i,j) \in A,\ i \in \{0\} \cup C,\ j \in C,\ \forall k \in K \tag{24}$$

$$0 \le W_j^k \le Q_k z_j^k, \qquad \forall j \in C,\ \forall k \in K \tag{25}$$

$$R_0^k = R_k^{\max} y_k, \qquad \forall k \in K \tag{26}$$

$$0 \le R_i^k \le R_k^{\max}, \qquad \forall i \in V,\ \forall k \in K \tag{27}$$

$$d_{ij} x_{ij}^k \le R_k^{\max}, \qquad \forall (i,j) \in A,\ \forall k \in K \tag{28}$$

$$y_i^k \le z_i^k, \qquad \forall i \in C,\ \forall k \in K \tag{29}$$

$$R_j^k = \begin{cases} R_i^k - d_{ij}, & x_{ij}^k = 1,\ y_i^k = 0,\ R_i^k \ge d_{ij}, \\ R_k^{\max} - d_{ij}, & x_{ij}^k = 1,\ y_i^k = 1, \end{cases} \quad \forall (i,j) \in A,\ i \in C,\ j \in C \cup \{n+1\},\ \forall k \in K \tag{30}$$

4. Methods

4.1. Overall Framework of the Proposed Method

To overcome the limitation of Euclidean-distance-based routing cost in complex 3D low-altitude environments, TeCoR-UAV adopts a two-stage topology-based cooperative planning framework. In the offline stage (Stage I), executable 3D flight trajectories between logistics nodes are precomputed and abstracted into a weighted topological graph. In the online stage (Stage II), this graph is reused for multi-UAV task allocation and route optimization, so that realistic traversal costs can directly participate in cooperative decision-making without repeatedly solving continuous-space obstacle-avoidance problems. The overall architecture is shown in Figure 3, and the main procedure is summarized in Algorithm 1.

Algorithm 1: TeCoR-UAV framework: overall procedure

Input: Physical environment model $\mathcal{M}$, node set V, heterogeneous UAV set K, task request set T, optimization hyperparameters par

Output: Task allocation and path planning solution S^*, corresponding trajectory Tr^*

```
if M is given for the first time or M changes then
    (G, Π) ← TopologyExtraction(M, V)
    // G = (V, E, w) is the weighted topological graph; it is complete
       in regular scenarios and pruned in topology-restricted
       scenarios
    // Π denotes the set of precomputed collision-free 3D
       trajectories associated with graph edges
while T ≠ ∅ do
    (S*, Tr*) ← HierarchicalSolver(T, K, G, Π, par)
    // HierarchicalSolver(·) is described in Algorithm 2
    print(S*, Tr*)
return S*, Tr*
```

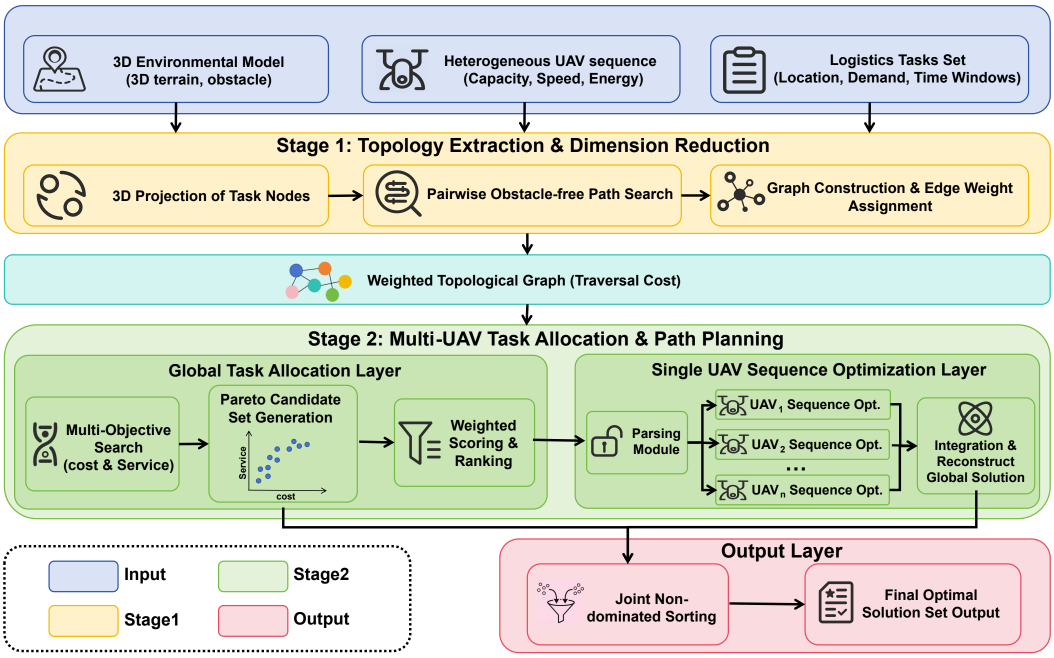

Figure 3. Schematic diagram of the TeCoR-UAV Optimization Framework.

The Algorithm 1 takes the physical environment model $\mathcal{M}$, the node set V, the heterogeneous UAV fleet K, the task set T and the set of algorithmic parameters par as input, and outputs the set of collaborative routing solution S^* together with the corresponding 3D trajectories Tr^*. In Stage I, $\mathcal{M}$ and V are used to construct a weighted topological graph $G = (V, E, w)$ and a precomputed trajectory set Π. The graph G enables rapid online queries of the realistic traversal costs between nodes, while Π stores the collision-free flight path associated with each edge. In Stage II, (G, Π) is reused as an environment-aware input, and once task requests are generated at the depot, the problem is solved through a hierarchical solver. The lower-level global task allocation layer generates candidate task-assignment structures and coarse visiting orders, whereas the upper-level single-UAV refinement layer further optimizes the execution sequence of each UAV under fixed assignment structures. Throughout the evolutionary process, fitness evaluation is performed directly using G, from which the temporal, energy, and service-related metrics are derived. Finally, by jointly applying non-dominated filtering across the outcomes of both layers, the algorithm yields a Pareto solution set that balances assignment efficiency and operational effectiveness, together with the corresponding 3D flight trajectories.

Algorithm 2: Stage-II hierarchical genetic algorithm for collaborative route planning

Input: Task request set T; heterogeneous UAV set K; weighted topology graph G generated in Stage I; collision-free 3D trajectory set Π; algorithm hyperparameter set par

Output: Optimal solution S^*; corresponding trajectory Tr^*

// par contains the population sizes N_l and N_h, maximum evolution generations $G^l_{\max}$ and $G^h_{\max}$, crossover probabilities p^l_c and p^h_c, mutation probabilities p^l_m and p^h_m, and the feasibility-checking limit $n_{\max}$

$Pop_{low} \leftarrow$ MixInitializePopulation(T, G, K, par)
$Pop_{low} \leftarrow$ LowerLayerEvolution(Pop_{low}, G, par)
$ParetoFront_l \leftarrow$ ExtractParetoSet(Pop_{low})
$rep \leftarrow$ SelectRepresentative$(ParetoFront_l)$
$TaskSeqs \leftarrow$ SplitByUAV(rep) // $[[tasks\ of\ UAV_1], \ldots, [tasks\ of\ UAV_n]]$
$ParetoFront^{all}_h \leftarrow \emptyset$
foreach *seq in TaskSeqs* **do**
  $Pop_h \leftarrow$ InitializeUpperLayer(seq)
  $ParetoFront_h \leftarrow$ HigherLayerEvolve(Pop_h, G, par)
  Append$(ParetoFront^{all}_h, ParetoFront_h)$
$AllParetoFront_h \leftarrow$ IntegrateSolutions$(ParetoFront^{all}_h)$
$S^* \leftarrow$ GlobalSelection$(ParetoFront_l, AllParetoFront_h)$
$\mathrm{Tr}^* \leftarrow$ ReconstructTrajectories(S^*, Π)
return S^*, Tr^*

4.2. Topology Extraction

To overcome the inability of conventional Euclidean distances to represent true flight cost, we precompute executable routes between nodes before collaborative routing and abstract the resulting connectivity into a weighted topological graph. Since the locations of depots and service points in logistics delivery networks typically remain stable over relatively long time horizons, the reachability relations and traversal costs between nodes can be computed in advance for the currently available airspace information and then reused online in graph form during route optimization.

The topology extraction procedure takes the 3D airspace model $\mathcal{M}$ and the node set V as input. Environmental constraints are jointly characterized by a terrain elevation field, a set of discrete obstacles $\mathcal{O}$, and known restrictions on airspace. In mountain scenarios, each flight trajectory must remain above the local terrain surface; in urban environments, it must maintain a minimum safety clearance δ from obstacles.

For an arbitrary node pair (v_i, v_j), we construct a safe and smooth executable 3D route π_{ij} through polyline path search, continuous trajectory smoothing, and sample-based feasibility correction. This procedure is implemented using particle swarm optimization and the generation of B-spline curves. On this basis, an undirected weighted graph

$G = (V, E, w)$ is constructed over all nodes. In regular scenarios, G is complete; in topology-restricted scenarios, selected inaccessible edges are removed from the complete initial graph. The set of precomputed trajectories is defined as $\Pi = \{\pi_{ij} \mid (v_i, v_j) \in E\}$. To ensure consistent cost representation in online planning, the edge weight w_{ij} is defined as the arc length sampled of π_{ij}.

Since topology extraction is conducted before collaborative routing, known changes in airspace restrictions can be incorporated before mission execution. When temporary no-fly zones or updated access rules are announced, affected edges are revalidated under the updated airspace constraints. Edges intersecting newly restricted regions are removed from the topological graph, while reachable node pairs requiring different detours are locally recomputed and assigned updated edge weights. The updated topological graph and trajectory set are then used for subsequent routing optimization.

This stage outputs the weighted topological graph G and the trajectory set Π for reuse in Stage II. The collaborative planner operates directly on G, using edge weights as traversal costs for task allocation, route composition, and fitness evaluation. After a graph-level route is obtained, the corresponding physical trajectory is reconstructed by concatenating the route segments retrieved from Π.

4.3. Collaborative Route Planning Stage

In low-altitude logistics delivery scenarios, collaborative multi-UAV planning must jointly address task allocation and route optimization under payload, energy, UAV range, time windows, and heterogeneous vehicle capabilities. To tackle the excessive complexity of the unified search space and the difficulty of generating feasible solutions directly, we developed a hierarchical collaborative optimization strategy structured as lower-level global allocation and upper-level local refinement. Operating on the weighted topological graph G, the lower level generates feasible assignment schemes by jointly considering task demand, spatial configuration, and time-window constraints, while balancing delivery cost against service quality. With task assignments fixed, the upper level further refines the task execution sequence of each individual UAV and reconstructs the corresponding physical trajectories using the trajectory set Π. This hierarchical design helps control computational complexity while preserving both global feasibility and local execution efficiency. The complete procedure is presented in Algorithm 2.

The Algorithm 2 takes the task request set T, the heterogeneous UAV set K, the weighted topological graph G, the trajectory set Π and the algorithmic parameter set par as input, and outputs the collaborative solution set S^* together with the corresponding 3D trajectories Tr^*. Lines 1–3 complete the lower-level global allocation process. Specifically, a hybrid initialization strategy generates the initial population Pop_{low}, and then an evolutionary search is conducted to obtain the lower-level Pareto candidate set $ParetoFront_l$. In line 4, a representative solution rep is selected from $ParetoFront_l$ using a preference-based weighted scoring strategy, which only determines the entry solution for upper-level refinement and does not replace the final non-dominated evaluation. In lines 5–10, rep is decoded into subsequences of specific tasks for UAV according to the chromosome encoding rule, and each subsequence is refined independently in the upper level to produce single Pareto sets for UAVs. Lines 11–12 integrate these upper-level results into complete collaborative candidates and jointly evaluate them with $ParetoFront_l$ to determine the final set of solutions S^*. Finally, the set of trajectories Π is used to map the selected graph-level routes back to executable 3D flight trajectories Tr^*.

4.3.1. Solution Encoding Scheme

To jointly represent task assignment and execution order in collaborative multi-UAV planning, we adopt a separator-based one-dimensional integer encoding scheme for lower-level candidate solutions. After decoding, the chromosome directly yields the task subsequence assigned to each UAV, providing the interface between lower-level global allocation and upper-level local refinement. The upper level then refines only the internal order of each single-UAV task subsequence while keeping the task assignments fixed. The encoding is defined as follows.

Definition 1 (Encoding scheme)**.** *Given m heterogeneous UAVs and n delivery tasks $T = \{t_1, t_2, \ldots, t_n\}$, define the set of task indexes as $T_c = \{1, 2, \ldots, n\}$ and the set of separators as $D_f = \{n+1, \ldots, n+m-1\}$. A lower-level chromosome is represented as an integer sequence $X = [x_1, x_2, \ldots, x_L]$ of length $L = n + m - 1$, where $x_i \in T_c$ denotes a task index and $x_i \in D_f$ denotes a separator. Separators partition X into m ordered task segments, each corresponding to one UAV; the r-th segment represents the execution sequence of the UAV r. Consecutive separators are allowed, indicating that the corresponding UAV is not assigned any task. The encoding satisfies three basic constraints: each task index appears exactly once; the chromosome contains exactly $m-1$ separators, i.e., $\sum_{i=1}^{L} \mathbf{1}[x_i \in D_f] = m - 1$; and every element belongs to $T_c \cup D_f = \{1, 2, \ldots, n+m-1\}$.*

Figure 4 illustrates the proposed encoding scheme. For example, when $m = 3$ and $n = 8$, the separators are 9 and 10; thus, $X = [7, 5, 1, 9, 3, 2, 10, 4, 6, 8]$ assigns tasks $[7, 5, 1]$, $[3, 2]$, and $[4, 6, 8]$ to UAVs u_1, u_2 and u_3, respectively.

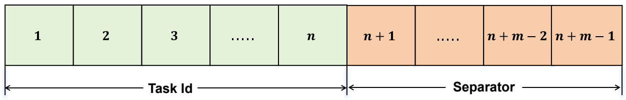

Figure 4. Schematic diagram of the Lower-Layer Chromosome Encoding Method.

4.3.2. Lower-Level Global Allocation

The lower level aims to generate globally feasible task-allocation candidates on the topological graph G while determining a coarse-grained execution order for all tasks. Because strongly coupled constraints make the feasible region sparse, random initialization alone is unlikely to provide high-quality feasible individuals. To improve adaptability to different spatial task distributions and heterogeneous resource-allocation patterns, four initialization strategies are combined in equal proportions to build the lower-level initial population. During evolution, a cyclic feasibility-checking and resampling step is used to prevent infeasible offspring from propagating in the population.

The four initialization strategies are designed to provide complementary feasible structures from spatial proximity, regional continuity, temporal priority, and demand–capacity matching, respectively.

(1) Cluster-based initialization. The service points are first grouped by K-means according to spatial proximity. Starting from the cluster containing the depot, a cluster-level visiting order is generated using a nearest-neighbor rule. Then a global task sequence is formed following this cluster order, with random permutation within each cluster to preserve diversity. The sequence is subsequently partitioned according to UAV payload limits to complete task assignment and chromosome construction, thereby introducing spatial locality into the initial population.

(2) Spatially adaptive partition-based initialization strategy. This strategy improves regional continuity by adaptively defining the coordinate origin and spatial traversal order according to the depot location in the task space, as illustrated in Figure 5. When the depot is located near the boundary region, as shown in Figure 5b, the geometric center of the task space is taken as the origin, and the quadrants are traversed counterclockwise starting from the quadrant containing the depot. When the depot lies in the interior core region, as shown in Figure 5c, the depot itself is used as the origin, and the traversal begins from the first quadrant counterclockwise. Tasks within each quadrant are randomly permuted and concatenated according to the quadrant sequence to form a global task sequence, which is then assigned and encoded under UAV payload and route-feasibility constraints. This strategy introduces coarse regional continuity to the initial population.

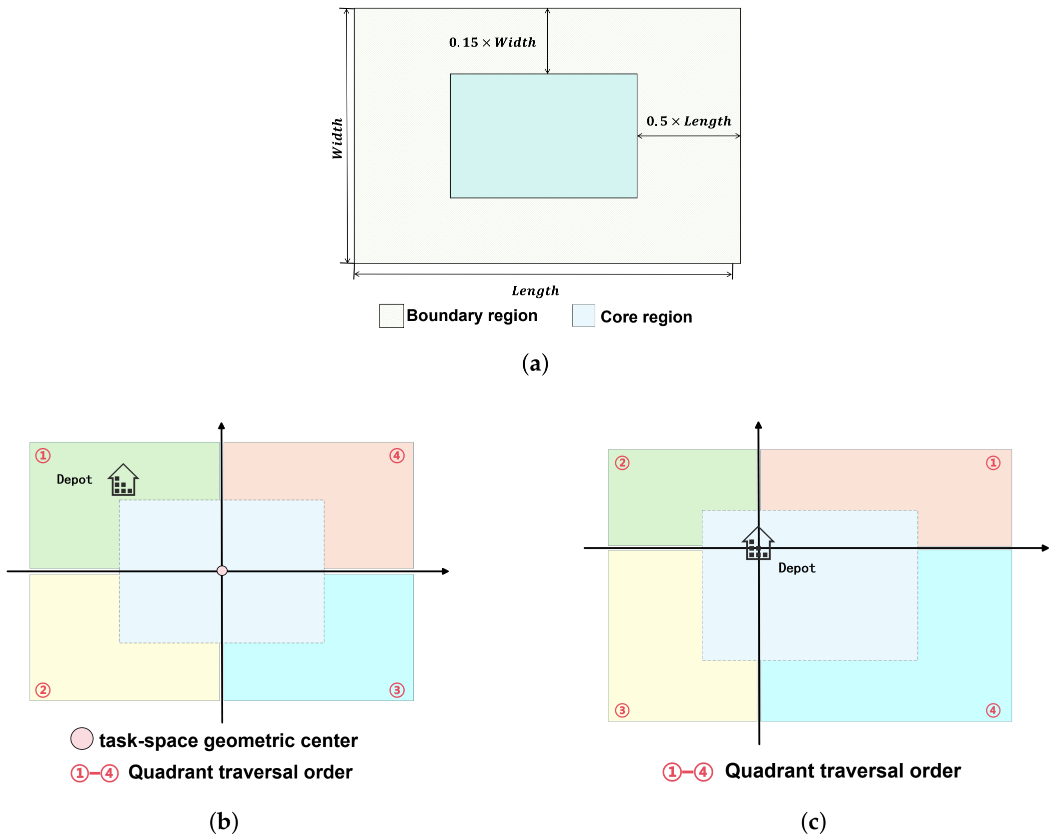

Figure 5. Spatially adaptive partition-based initialization: (**a**) task-space boundary/core partition; (**b**) quadrant traversal when the depot is located in the boundary region; (**c**) quadrant traversal when the depot is located in the core region.

(3) Initialization based on FIFO. Tasks are sorted by arrival time and assigned to the UAV sequentially according to the UAV index. For each task, the current UAV is selected if the assignment satisfies the payload and route-feasibility constraints; otherwise, the next feasible UAV is considered. After the assignment, the task sequence of each UAV is randomly permuted to generate multiple initial chromosomes.

(4) Best-fit allocation-based initialization. This strategy emphasizes the compatibility between task demand and UAV capability. After tasks and UAVs are jointly ranked according to payload requirements and platform capacity, they are sequentially assigned to suitable UAVs to improve demand–capability matching. The resulting task subsequences are then randomly permuted to produce multiple initial chromosomes.

During iterative optimization, the lower-level population is updated by binary tournament selection, partially mapped crossover, and inversion mutation. Each newly generated offspring is then checked against assignment-structure, payload, UAV-range, energy-consumption, and resource-matching constraints. If the offspring is infeasible, the affected operator positions are randomly resampled, and the corresponding crossover or mutation is repeated until a feasible offspring is obtained or the maximum number of attempts $n_{\max}$ is reached. If this limit is reached, the infeasible offspring is discarded, and the fitter's feasible parent is retained. This feasibility check and resampling step prevent infeasible individuals from propagating during lower-level evolution.

4.3.3. Upper-Level Local Refinement and Inter-Level Information Transfer

The upper-level procedure consists of four steps: representative solution selection, upper-level population construction, local refinement of a single UAV, and joint reconstruction and filtering. The central idea is to select, from the lower-level Pareto set, a representative solution that is consistent with the current task preference, refine each UAV sequence locally while preserving task assignments, and then reconstruct complete collaborative solutions for final joint filtering.

In this framework, the transfer of information between layers is implemented through structural inheritance and the final integration of the set of candidates. The lower-level Pareto front is retained as the global candidate archive to preserve diverse trade-off solutions formed by different task-assignment structures and coarse-grained visiting orders. The upper-level refinement is preference-oriented and optimizes only the single-UAV execution sequences of the representative solution selected from the lower-level Pareto set, aiming to improve its execution quality rather than replace the entire lower-level Pareto front. Therefore, the refined upper-level candidates are treated as supplementary solutions and are jointly filtered with the lower-level Pareto candidates during final non-dominated selection. This design preserves Pareto-front diversity while introducing local refinement for executable preference-related solutions, thereby balancing global trade-off preservation and preference-oriented solution improvement.

(1) Selection of Representative-solutions. To provide a preference-related entry for the upper-level refinement, a representative solution *rep* is selected from $ParetoFront_l$ rather than passing the entire lower-level Pareto set to the upper level. Selection is performed using a weighted scoring rule based on the task preference ratio $\vartheta \in [0, 1]$. Since cost is minimized, while service quality is maximized, the two objectives are first normalized and then aggregated, as shown in Equation (31), where $score_i$ denotes the preference score of the i -th lower-level candidate, while $cost_i^{norm}$ and sa_i^{norm} denote normalized cost and service quality indicators, respectively. The lower-level candidate with the lowest $score_i$ is used as the representative solution *rep* for upper-level refinement.

$$score_i = cost_i^{norm} \cdot \vartheta - sa_i^{norm} \cdot (1 - \vartheta) \tag{31}$$

(2) Construction of the Upper-level population. The representative solution *rep* is first decoded into UAV-specific task subsequences. The upper level then constructs its population directly from these subsequences without redefining any cross-UAV encoding. For each UAV, the upper-level population consists of the inherited task order from *rep* and randomly permuted variants of the same subsequence. Since task assignments are fixed, these permutations expand the local search space without changing the allocation structure.

(3) Local Single-UAV refinement. Upper-level optimization is performed independently for each UAV by refining only the internal execution order of its assigned tasks. Since task assignments remain unchanged, the payload relationships and resource-matching structure determined at the lower level are preserved. Meanwhile, time windows are treated as soft constraints, with violations reflected through objective penalties. For this reason, the feasibility-repair mechanism used in the lower level is no longer required. Instead, fitness evaluation directly guides the local search for improved trade-offs among travel time, energy consumption, and service quality. All evaluations are conducted on the topological graph G, yielding a Pareto set for each UAV.

(4) Alignment, reconstruction, and joint filtering. Because the sizes of the single-UAV Pareto sets may differ, exhaustive combinatorial reconstruction would be computationally expensive. Therefore, a score-guided alignment strategy is adopted. Let $Size_{Max}$ denote the maximum cardinality among all single-UAV Pareto sets. Each smaller set is padded by replicating its best-scoring solution according to Equation (31) until all sets reach $Size_{Max}$. Solutions with matched indices are then concatenated and re-encoded to reconstruct complete upper-level collaborative candidates. Finally, these upper-level candidates are jointly filtered together with $ParetoFront_l$ through non-dominated sorting, with duplicate and dominated solutions removed to obtain the final collaborative planning set S^*. Using the set of trajectories Π generated in Stage I, the graph-level paths in S^* are then mapped to executable 3D flight trajectories Tr^*.

4.4. Computational Complexity Analysis

To clarify the computational burden of TeCoR-UAV, we briefly analyze the main computational costs of the two-stage framework. Let n denote the number of service points, m denote the number of UAVs, and $|E|$ denote the number of feasible edges in the extracted topological graph. In Stage I, executable 3D trajectories are precomputed for graph edges. If N_{PSO}, G_{PSO}, and S denote the number of particles, the maximum number of PSO iterations, and the number of trajectory-evaluation samples, respectively, the complexity of the preprocessing can be approximated as $O(|E|N_{\mathrm{PSO}}G_{\mathrm{PSO}}S)$. This cost is incurred before collaborative routing, and the resulting weighted graph and trajectory set are reused in Stage II.

In Stage II, the lower-level evolutionary search has a population size N_l and runs for $G_l^{\max}$ generations. Considering route evaluation, feasibility repair, and non-dominated sorting, its complexity can be approximated as $O\big(G_l^{\max}(N_l n_{\max} n + N_l^2)\big)$. The upper-level refinement uses a population size N_h and runs for $G_h^{\max}$ generations under fixed task assignments, and its complexity is approximated as $O\big(G_h^{\max}(N_h n + m N_h^2)\big)$.

Therefore, Stage I introduces an offline preprocessing cost, whereas Stage II avoids repeated continuous-space 3D path searches by evaluating candidate routes directly on the precomputed topological graph. The empirical runtime performance is further reported in the experimental section.

5. Simulation Experiments and Results Analysis

To systematically evaluate the effectiveness and applicability of TeCoR-UAV across various scenarios, this section presents a comprehensive set of simulation experiments conducted under varying environmental and operational conditions. The experimental study is organized around the following three research questions:

RQ1: Can TeCoR-UAV generate feasible and executable cooperative delivery plans in obstacle-constrained 3D environments?

RQ2: Under different physical environments, task scales, and UAV fleet configurations, does TeCoR-UAV outperform existing methods in terms of solution optimality and diversity in multi-objective optimization?

RQ3: Within the collaborative route planning stage (Stage II), how do the key mechanisms proposed in this study individually contribute to the overall improvement in the performance of the solution?

In addition, beyond the scope of the above research questions, we further perform supplementary sensitivity analyses on key hyperparameters, the preference coefficient, and fleet resource availability.

5.1. Experimental Settings

5.1.1. UAV Fleet Configuration

To characterize heterogeneity in payload capacity, flight efficiency, endurance, and operating cost, we consider three categories of UAV: light, medium, and heavy-duty platforms. To preserve physical realism in simulation experiments, the key technical specifications of these UAVs are defined with reference to publicly available parameter ranges of representative UAVs for commercial logistics [42,43] and further harmonized to match the scenarios considered in this study. The main parameters include maximum payload, unloaded cruising speed, maximum full-battery flight range, fixed activation cost, baseline energy-consumption coefficient per unit distance, and maintenance cost, as summarized in Table 2. Here, the maximum full-battery flight range $R_k^{\max}$ denotes the maximum cumulative flight distance that the UAV k can complete under full-battery conditions. The fixed activation cost denotes the one-time cost incurred when a UAV is deployed, whereas the maintenance cost represents the support cost accumulated over the execution cycle. These parameters are fully consistent with the payload–speed and payload–energy models introduced earlier, and are used to capture the heterogeneous operational capabilities of different UAV types in delivery missions.

Table 2. UAV types and operating parameters.

Type	Capacity Q (kg)	Speed V_0 (km/h)	Range $R_k^{\max}$ (km)	Fixed Cost f_k (¥)	Energy e_0 (per 10 km)	Maintenance Cost ma_k (¥)
W1	8	35	35	16	0.30	10
W2	13	30	30	26	0.45	20
W3	16	20	25	36	0.65	30

5.1.2. Task Scenario Configuration

Based on the above fleet settings, we further construct task scenarios under different environmental and demand conditions to evaluate the applicability of the proposed algorithm at varying levels of operational complexity. Specifically, two representative 3D test environments are considered, namely mountainous and urban scenarios, and instances with 10, 30, and 50 task nodes are generated to represent small, medium, and large-scale low-altitude logistics demands, respectively. For each scenario, a single depot is specified and paired with a heterogeneous UAV fleet of appropriate size. As the task scale increases, the fleet size is expanded accordingly, with a higher proportion of medium- and heavy-duty UAVs introduced to reflect the increased payload requirements of large-scale delivery operations. Task nodes are generated randomly within the admissible flight space. The associated service requests include three types: delivery, pickup, and integrated delivery–pickup, and all tasks are subject to service time-window constraints. Detailed scenario categories, task scales, and UAV fleet configurations are summarized in Table 3. Due to space limitations, the full data for node coordinates, logistics demand quantities, and time-window parameters are provided in the Supplementary Dataset S1.

Table 3. Benchmark scenario settings and heterogeneous UAV fleet compositions.

Scenario	Environment	Number of Tasks	Number of UAVs		
			W1	W2	W3
S1	Mountainous	10	1	1	1
S2	Mountainous	30	2	1	2
S3	Mountainous	50	1	3	3
S4	Urban	10	1	1	1
S5	Urban	30	2	1	2
S6	Urban	50	1	3	3

5.1.3. Algorithmic and Parameter Settings

Given the above fleet and scenario configurations, we next specify the optimization parameters used in the experiments. The Stage I topology-extraction module is treated as a fixed environmental preprocessing procedure, and its parameters are kept unchanged for all scenarios; detailed settings are reported in Appendix A, Table A1 for reproducibility. Consequently, this subsection reports only the hyperparameter settings associated with Stage II, including population size, maximum number of generations, crossover probability, mutation probability, feasibility-repair limit, and preference coefficient, as summarized in Table 4. In particular, the preference coefficient ϑ is set to 0.5, indicating equal emphasis on cost control and service quality. The remaining parameters related to the model are fixed according to the formulations introduced in Section 3, with $\alpha = 0.1$, $\beta = 0.01$, $\rho = 0.5$, and $\tau = 3$. In addition, an early stopping criterion is adopted in the evolutionary process. If the HV value does not improve for 10 consecutive generations, the corresponding evolutionary search ends early; otherwise, it stops when the maximum number of generations is reached.

Table 4. Stage-II algorithmic parameter settings.

Category	Parameter	Symbol	Value
Lower-level genetic algorithm parameters	Population size	N_l	50
	Maximum number of generations	$G_l^{\max}$	100
	Crossover probability	p_c^l	0.8
	Mutation probability	p_m^l	0.1
Upper-level genetic algorithm parameters	Population size	N_h	30
	Maximum number of generations	$G_h^{\max}$	20
	Crossover probability	p_c^h	0.8
	Mutation probability	p_m^h	0.3
Other optimization parameters	Feasibility-repair limit	$n_{\max}$	10
	Preference coefficient	ϑ	0.5

5.1.4. Evaluation Metrics

The algorithms are evaluated from three perspectives: representative-solution performance, Pareto-set quality, and computational efficiency. For multi-objective methods, a unified normalization scale is applied to the Pareto candidates obtained on the same instance, and the weighted scoring rule in Equation (31) is used to select a representative solution for reporting cost and service quality. Single-objective baselines are evaluated under the same task-preference setting to ensure comparability. In addition, the actual running time of each algorithm is recorded in the same experimental environment to evaluate computational efficiency.

The hypervolume (HV) indicator is employed to evaluate the quality of multi-objective solution sets. HV jointly reflects both the convergence and distributional diversity of the Pareto front and is computed only for algorithms capable of producing multi-objective outputs. A larger HV value indicates that the obtained solution set covers a broader region of the objective space with a more desirable distribution, and therefore corresponds to better overall optimization performance. The HV calculation is given in Equation (32), where $f_d(x)$ denotes the value of the solution x on the d-th objective, $\mathrm{Leb}(\cdot)$ denotes the Lebesgue measure used to compute the volume, and $z^{ref} = (z_1^{ref}, z_2^{ref}, \ldots, z_d^{ref})$ denotes the reference point. In this study, the reference point is constructed for each test instance from the worst objective values attained by all algorithms along each objective dimension.

$$HV(P) = \mathrm{Leb}\left(\bigcup_{x \in P}\left[f_1(x), z_1^{ref}\right] \times \cdots \times \left[f_d(x), z_d^{ref}\right]\right) \tag{32}$$

5.2. Feasibility Verification of TeCoR-UAV

To answer RQ1, this subsection verifies whether TeCoR-UAV can generate feasible cooperative delivery solutions for low-altitude logistics scenarios with 3D obstacles. Specifically, the proposed algorithm is applied to the six task scenarios summarized in Table 3, including different types of environments, task scales, and UAV fleet configurations. For each scenario, the final representative solution obtained by TeCoR-UAV is mapped back to the corresponding 3D obstacle-aware trajectories and visualized in Figure 6.

As shown in Figure 6, TeCoR-UAV can generate feasible cooperative delivery plans for all six scenarios tested. The planned routes connect the depot and the task nodes assigned to different UAVs while avoiding obstacle-occupied regions. The visualization results show that the proposed method is applicable to both mountainous and urban environments and remains feasible on different task scales. Therefore, TeCoR-UAV can transform high-level cooperative routing decisions into executable obstacle-aware flight trajectories, which verifies the feasibility of the proposed method for the considered low-altitude logistics problem.

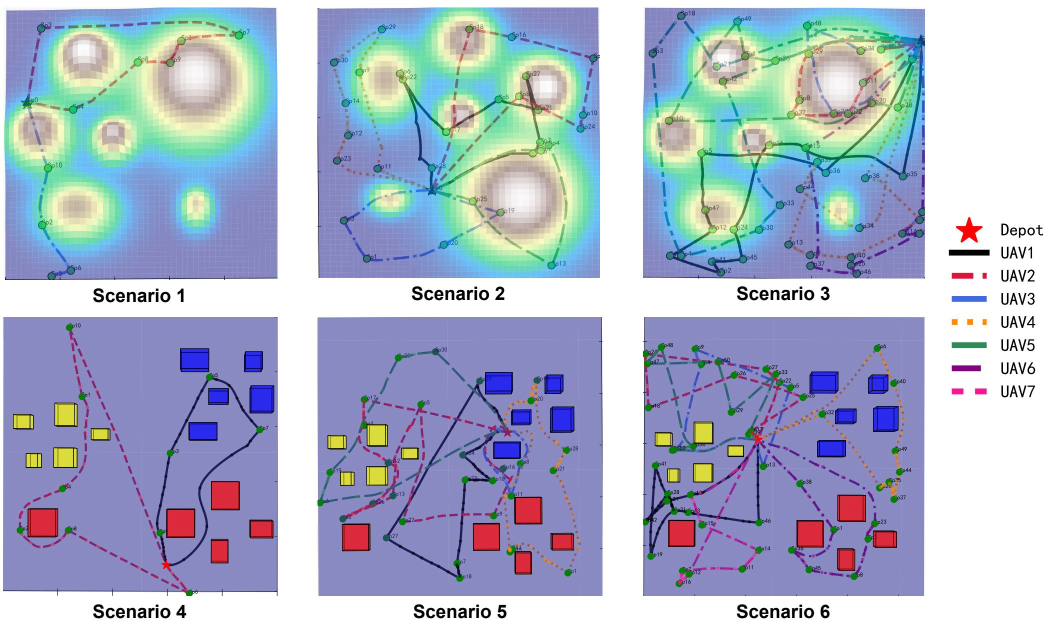

Figure 6. Visualization of feasible cooperative delivery trajectories generated by TeCoR-UAV in six obstacle-constrained scenarios. Colored boxes denote obstacle blocks in urban scenarios, and different line colors indicate routes assigned to different UAVs.

5.3. Comparative Analysis of Collaborative Route Planning Performance

To evaluate the overall performance of TeCoR-UAV, comparative experiments were conducted on six benchmark scenarios (S1–S6) and one restricted-topology scenario (S-R). The S-R scenario was constructed by removing a subset of inter-node connections from the initial complete graph, resulting in a weakly connected topology with 25 task nodes and 215 retained edges. Its fleet consists of one W1 UAV, two W2 UAVs, and one W3 UAV.

Comparison baselines include NSGA-II [44], NSGA-II-VRPD [27], SAVND [28], a reference method based on DP-MILP decomposition, GAPPO-LS [34], and D-LOF [40].

NSGA-II and NSGA-II-VRPD are used as representative evolutionary and vehicle-routing-based multi-objective baselines, respectively, while SAVND provides a single-solution neighborhood-search comparison. DP-MILP is used only as an exact reference for the simplified small-scale scalarized route-selection problem, where feasible single-UAV routes are first enumerated by dynamic programming and then selected through a MILP master problem under a normalized weighted-sum objective. GAPPO-LS is introduced to examine the applicability of reinforcement-learning-based task allocation followed by local search. D-LOF is included as a representative online path-feedback baseline, in which obstacle-aware path planning is invoked during the task-allocation evolution process, and the resulting flight cost is fed back into fitness evaluation. For fairness, TeCoR-UAV and graph-based comparison methods use the same obstacle-aware weighted topology generated by Stage I as the shared path-cost basis, while D-LOF retains its original online path-planning mechanism and does not use the precomputed Stage I edge weights. Since S-R is specifically constructed to evaluate graph-based planning under restricted topology, it is not used for comparison with D-LOF.

Table 5 and Figure 7 compare the performance of the representative solution, the quality of the Pareto-set, and the computational efficiency of the algorithms tested in all scenarios. In the small-scale scenarios S1 and S4, most methods obtain feasible solutions with relatively close representative performance, suggesting that the constraint coupling is still manageable when the number of tasks is limited. D-LOF also obtains feasible solutions in these scenarios, indicating that online path-feedback planning can be effective in small instances. However, as the task scale increases, the advantage of TeCoR-UAV becomes more evident. In S2, S3, S5, S6, and S-R, TeCoR-UAV generally achieves a more balanced cost–service quality trade-off and higher HV values among the multi-objective algorithms, demonstrating stronger solution quality and Pareto-set quality under more complex payload, time-window, endurance, and topological constraints.

Table 5. Performance comparison of TeCoR-UAV and comparison methods under different scenarios.

Scenario	Indicator	TeCoR-UAV	NSGA-II [44]	NSGA-II-VRPD [27]	SAVND [28]	DP-MILP	GAPPO-LS [34]	D-LOF [40]
S1	Average	[81.308, 90.6%]	[88.897, 92.6%]	[80.377, 92.7%]	[94.200, 95.6%]	[83.841, 100.0%]	[93.954, 99.0%]	[85.194, 97.7%]
	Average HV	41.382	41.885	41.887	–	–	–	–
	Time (min)	0.807(2.145)	0.220(2.145)	0.440(2.145)	6.132(2.145)	0.182(2.145)	1.881(2.145)	44.484
S2	Average	[258.025, 80.5%]	[274.878, 55.3%]	[269.144, 81.1%]	[249.431, 45.1%]	–	[284.489, 74.4%]	[264.690, 85.8%]
	Average HV	15.636	3.312	11.711	–	–	–	–
	Time (min)	3.044(19.094)	0.060(19.094)	1.956(19.094)	5.303(19.094)	–	23.347(19.094)	115.851
S3	Average	[420.902, 74.1%]	–	[437.926, 68.8%]	[445.369, 48.6%]	–	[466.664, 55.7%]	–
	Average HV	14.308	–	7.110	–	–	–	–
	Time (min)	9.701(55.758)	–	8.072(55.758)	6.486(55.758)	–	249.618(55.758)	–
S4	Average	[81.415, 99.2%]	[81.605, 95.6%]	[81.402, 94.4%]	[84.833, 98.8%]	[82.743, 99.9%]	[99.525, 99.6%]	[81.386, 96.1%]
	Average HV	32.561	32.637	32.619	–	–	–	–
	Time (min)	0.796(2.636)	0.318(2.636)	0.603(2.636)	6.065(2.636)	0.147(2.636)	1.475(2.636)	50.199
S5	Average	[257.126, 82.8%]	[273.616, 53.1%]	[260.116, 77.6%]	[246.018, 43.3%]	–	[271.070, 69.9%]	[265.103, 88.4%]
	Average HV	12.151	3.351	8.608	–	–	–	–
	Time (min)	4.056(21.393)	0.085(21.393)	3.096(21.393)	7.834(21.393)	–	23.161(21.393)	114.346
S6	Average	[425.744, 84.2%]	–	[437.957, 69.6%]	[438.060, 48.9%]	–	[454.351, 58.5%]	–
	Average HV	9.741	–	3.419	–	–	–	–
	Time (min)	9.206(57.692)	–	7.841(57.692)	8.601(57.692)	–	250.455(57.692)	–
S-R	Average	[214.398, 81.6%]	[219.421, 60.3%]	[217.862, 76.2%]	[225.482, 58.7%]	–	[225.129, 76.5%]	–
	Average HV	11.361	5.420	8.794	–	–	–	–
	Time (min)	2.570	0.217	1.485	10.393	–	8.402	–

Note: "–" indicates that the corresponding algorithm did not obtain a feasible solution, exceeded the predefined time budget, or that the metric is not applicable. For graph-based algorithms, the value in parentheses in the Time row denotes the offline Stage I topology-extraction time, while the value outside parentheses denotes the route-optimization time. Time is reported in minutes.

Compared with NSGA-II and NSGA-II-VRPD, TeCoR-UAV shows a stronger feasible-solution search capability and Pareto-set maintenance ability in complex scenarios. NSGA-II fails to obtain feasible solutions in some large-scale scenarios, while NSGA-II-VRPD, although adopting a vehicle-routing-oriented solution structure, still yields lower HV values than TeCoR-UAV in medium- and large-scale cases. SAVND obtains lower-cost solutions in some scenarios, but its service quality is generally limited, suggesting that a single-solution neighborhood search tends to favor local cost reduction at the expense of service satisfaction. GAPPO-LS generates feasible solutions in all scenarios; however, its cost and service-quality performance in large-scale cases remains weaker than that of TeCoR-UAV. This indicates that learning-based task allocation followed by local search is still insufficient to fully address the strongly coupled constraints considered in this study.

In addition, the D-LOF results highlight the computational advantage of topology reuse. D-LOF obtains feasible solutions in S1, S2, S4, and S5, where the number of task nodes is relatively limited and online A*-based evaluation can still be completed within the predefined time budget, albeit with a high computational cost. However, it exceeds the time budget in S3 and S6 because each fitness evaluation requires online obstacle-aware path planning, leading to an excessive number of repeated A* searches during evolution. By contrast, TeCoR-UAV performs obstacle-aware trajectory generation offline in Stage I and reuses the resulting weighted topology during Stage II, thus avoiding repeated online path searches while maintaining realistic flight-cost representation. Although TeCoR-UAV is not always the fastest method, it consistently generates feasible solutions in all tested scenarios and achieves a favorable balance between representative solution quality, Pareto-set quality, and computational efficiency.

To further verify the statistical reliability of the above differences, paired Wilcoxon signed-rank tests are performed on the ten-run cost and service-quality results with Bonferroni correction, and the adjusted p-values and Cohen's d_z effect sizes are reported in Table 6.

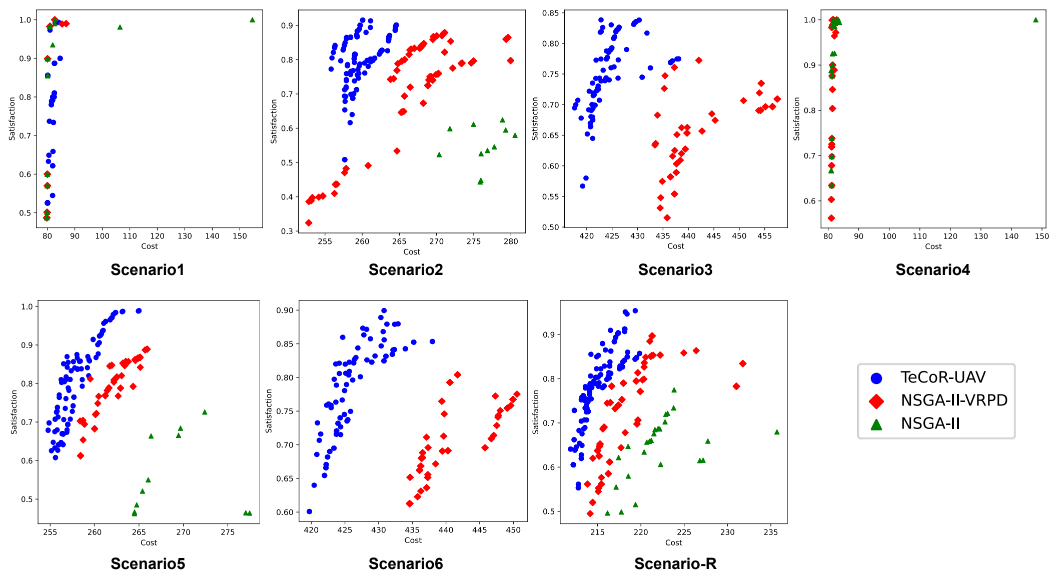

Figure 7. Pareto fronts of TeCoR-UAV and comparison methods under different scenarios.

The results in Table 6 further support the effectiveness of TeCoR-UAV. In large-scale scenarios such as S3 and S6, TeCoR-UAV significantly outperforms compared algorithms in both cost and satisfaction, indicating stronger optimization capability under highly coupled constraints. Several comparisons in S1 and S4 are not significant, which is consistent with the previous observation that the performance gap is less pronounced in small-scale scenarios. It should be noted that TeCoR-UAV significantly outperformed SAVND in the cost indicator in S2. However, TeCoR-UAV significantly outperforms SAVND in satisfaction in the same scenario, suggesting that SAVND achieves a lower cost at the expense of service quality. Therefore, this isolated cost advantage does not contradict the overall conclusion that TeCoR-UAV provides a more balanced cost–satisfaction trade-off.

Table 6. Statistical significance analysis between TeCoR-UAV and comparison algorithms.

Scenario	Indicator	TeCoR-UAV vs. NSGA-II-VRPD	TeCoR-UAV vs. SAVND	TeCoR-UAV vs. GAPPO-LS	TeCoR-UAV vs. D-LOF
S1	Cost	$(=, 1.000, 0.17)$	$(=, 0.059, 0.70)$	$(=, 0.059, 1.34)$	$(=, 0.059, 0.94)$
	Satisfaction	$(=, 1.000, 0.19)$	$(=, 1.000, -0.28)$	$(=, 0.117, -1.41)$	$(=, 1.000, 0.35)$
S2	Cost	$(\checkmark, 0.020, 1.16)$	$(\times, 0.020, -0.74)$	$(=, 1.000, -0.39)$	$(=, 1.000, -0.57)$
	Satisfaction	$(=, 0.195, 0.86)$	$(\checkmark, 0.020, 4.28)$	$(\checkmark, 0.020, 1.67)$	$(=, 0.098, -1.20)$
S3	Cost	$(\checkmark, 0.023, 2.89)$	$(\checkmark, 0.023, 5.89)$	$(\checkmark, 0.023, 8.14)$	$(--, --, --)$
	Satisfaction	$(\checkmark, 0.023, 1.88)$	$(\checkmark, 0.023, 7.95)$	$(\checkmark, 0.023, 5.38)$	$(--, --, --)$
S4	Cost	$(=, 1.000, -0.31)$	$(\checkmark, 0.020, 3.12)$	$(=, 0.039, 0.49)$	$(=, 1.000, -0.30)$
	Satisfaction	$(=, 1.000, -0.26)$	$(=, 1.000, -0.20)$	$(=, 1.000, -0.62)$	$(=, 1.000, 0.18)$
S5	Cost	$(=, 0.059, 1.19)$	$(=, 1.000, -0.48)$	$(\checkmark, 0.039, 1.76)$	$(\checkmark, 0.020, 7.85)$
	Satisfaction	$(=, 0.059, 1.24)$	$(\checkmark, 0.039, 3.74)$	$(\checkmark, 0.020, 2.20)$	$(=, 1.000, -0.43)$
S6	Cost	$(\checkmark, 0.023, 5.40)$	$(\checkmark, 0.023, 11.23)$	$(\checkmark, 0.023, 10.05)$	$(--, --, --)$
	Satisfaction	$(\checkmark, 0.023, 1.90)$	$(\checkmark, 0.023, 10.23)$	$(\checkmark, 0.023, 7.60)$	$(--, --, --)$
S-R	Cost	$(=, 0.273, 1.03)$	$(\checkmark, 0.039, 3.20)$	$(\checkmark, 0.039, 3.77)$	$(--, --, --)$
	Satisfaction	$(=, 1.000, 0.12)$	$(\checkmark, 0.039, 3.15)$	$(=, 1.000, 0.39)$	$(--, --, --)$

Note: Each tuple is reported as (s, p_{adj}, d_z), where s denotes the significance direction, p_{adj} is the Bonferroni-adjusted p-value obtained from the paired Wilcoxon signed-rank test, and d_z is Cohen's paired effect size. The symbols $\checkmark$, $\times$, and $=$ indicate that TeCoR-UAV is significantly better than, significantly worse than, and not significantly different from the compared algorithm, respectively. For Cost, lower values are better; for Satisfaction, higher values are better.

5.4. Ablation Analysis of Core Optimization Mechanisms

To further quantify the contribution of each key mechanism, we performed a progressive ablation study. Starting from a baseline configuration with lower-level optimization and random population initialization, we sequentially introduce the hybrid initialization strategy, the feasibility-repair mechanism, and the upper-level single-UAV sequence refinement mechanism, resulting in four variants: M4, TeCoR-UAV-c (lower-level global allocation + random population initialization); M3, TeCoR-UAV-b (M4 + hybrid initialization strategy); M2, TeCoR-UAV-a (M3 + feasibility-repair mechanism); and M1, TeCoR-UAV (M2 + upper-level single-UAV sequence refinement mechanism). Consequently, the comparison between M3 and M4 evaluates the contribution of hybrid initialization, the comparison between M2 and M3 isolates the effect of feasibility repair, and the comparison between M1 and M2 evaluates the benefit of the upper-level single-UAV sequence refinement. The evaluation metrics remain total operating cost, service quality, and hypervolume (HV), as defined in Section 5.1.4.

Table 7 and Figure 8 report the ablation results in all scenarios. This ablation study is conducted as an independent controlled experiment, where all variants are tested under the same scenario settings, parameter configurations, and computational budget. Therefore, the results are used mainly to compare the relative contributions of different components, rather than to duplicate the comparative results in Table 5. Overall, as the proposed mechanisms are progressively introduced, the complete TeCoR-UAV generally achieves better cost–service quality trade-offs and higher HV values in medium- and large-scale scenarios. The comparison between M3 and M4 highlights the contribution of the hybrid initialization strategy. Compared with purely random initialization, the hybrid strategy provides higher-quality initial populations and improves Pareto-set quality in more complex scenarios. For example, HV increases from 0.630 to 3.177 in S2 and from 0.306 to 4.484 in S5, indicating that hybrid initialization helps enlarge Pareto-front coverage and provides a stronger starting point for subsequent evolutionary search.

Table 7. Performance comparison of TeCoR-UAV and ablation variants under different scenarios.

Scenario	Indicator	TeCoR-UAV	TeCoR-UAV-a	TeCoR-UAV-b	TeCoR-UAV-c
S1	Average Average HV	[81.778, 86.3%] 3.697	[86.664, 94.1%] 3.697	[87.723, 98.2%] 3.418	[80.093, 89.9%] 5.031
S2	Average Average HV	[263.684, 87.7%] 4.599	[266.401, 83.2%] 3.566	[265.320, 75.7%] 3.177	[268.902, 59.9%] 0.630
S3	Average Average HV	[422.140, 83.0%] 2.422	[431.805, 87.2%] 2.045	[432.166, 85.7%] 1.568	– –
S4	Average Average HV	[81.448, 98.8%] 1.163	[81.695, 100.0%] 1.162	[84.074, 100.0%] 1.000	[82.488, 97.2%] 0.864
S5	Average Average HV	[257.601, 91.7%] 6.263	[262.066, 94.6%] 5.565	[258.529, 77.3%] 4.484	[268.800, 65.2%] 0.306
S6	Average Average HV	[400.024, 87.8%] 1.996	[405.720, 88.9%] 1.330	[406.252, 86.7%] 1.364	– –

Note: "–" indicates no feasible solution obtained by TeCoR-UAV-c.

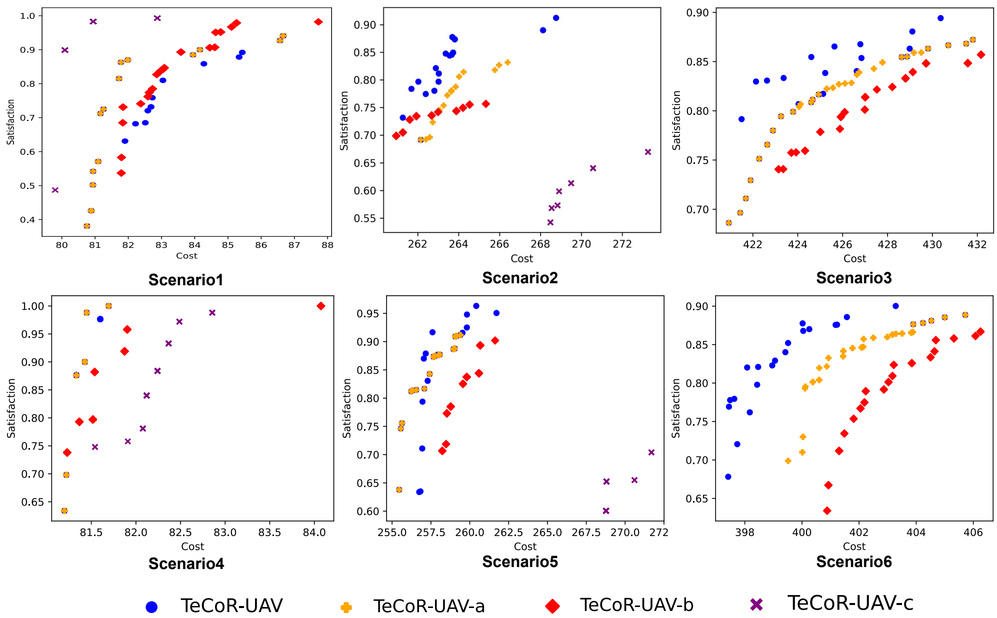

Figure 8. Pareto fronts of TeCoR-UAV and ablation variants under different scenarios.

The comparison between M2 and M3 shows the contribution of the feasibility-repair mechanism. Although the numerical gains in some scenarios are moderate, the Pareto fronts in Figure 8 show that the repaired variants generally produce more stable and better-distributed solution sets. This indicates that the feasibility-repair mechanism helps the evolutionary search remain within valid regions under coupled payload, endurance, and time-window constraints, thus improving the reliability of the search process in constrained scenarios.

The comparison between M1 and M2 reveals that the upper-level single-UAV sequence refinement mechanism yields gains that depend on the problem scale. In small-scale scenarios, the lower level can already identify reasonably effective visiting sequences, leaving limited room for further refinement. By contrast, in medium- and large-scale scenarios, the intra-route visiting order remains more difficult to optimize, and the upper-level refinement mechanism helps improve the final cost–service trade-off. As shown in Figure 8, the Pareto fronts obtained by the complete TeCoR-UAV are generally closer to the desirable upper-left region and exhibit more stable coverage in complex scenarios. This demonstrates the effectiveness of this mechanism in more complex task environments.

Some local deviations from the overall trend can still be observed. For example, in S1, the baseline variant M4 obtains the highest HV, suggesting that random initialization may retain stronger exploratory diversity in a small search space. In addition, TeCoR-UAV-a achieves slightly higher service quality than the complete TeCoR-UAV in some scenarios, such as S3, S5, and S6, but this is usually accompanied by higher operating costs or lower HV. These results indicate that individual mechanisms can improve a specific indicator in certain cases. Overall, however, the complete TeCoR-UAV provides a more balanced cost–service quality trade-off and more consistent Pareto-set performance, especially in medium- and large-scale scenarios.

5.5. Sensitivity Analysis of Key Parameters, Preference Coefficient, and Resource Availability

5.5.1. Sensitivity Analysis of Key Parameters

To justify parameter settings, a one-factor-at-a-time sensitivity analysis is conducted on five key lower-level parameters, including the population size N_l, the maximum number of generations $G_l^{\max}$, the crossover probability p_c^l, the mutation probability p_m^l, and the repair limit $n_{\max}$. The lower-level module directly determines the multi-UAV task-allocation structure and is responsible for the search for feasible solutions under strongly coupled constraints; therefore, its parameters have a more pronounced impact on overall performance. By contrast, the upper-level single-UAV sequence optimization performs local order refinement based on a given task-allocation result, where the search space is smaller and stable performance can be obtained using common genetic algorithm settings. Sensitivity experiments are conducted on the large-scale scenario S6, because larger instances involve stronger constraint coupling and a broader search space, making the influence of parameter settings on convergence behavior and solution set quality more evident. Candidate values are set as $N_l = \{50, 100, 150, 200\}$, $G_l^{\max} = \{50, 100, 150, 200\}$, $n_{\max} = \{1, 5, 10, 15, 20\}$, $p_c^l = \{0.4, 0.6, 0.8, 1.0\}$, and $p_m^l = \{0.05, 0.1, 0.2, 0.3\}$. Each parameter setting is independently tested three times, and the HV convergence curve and average computational time are recorded.

Figure 9 and Table 8 present the sensitivity results for the five key parameters. For this parameter-sensitivity experiment, the HV values in Figure 9 are calculated using the same fixed reference point $[1000, 0.0]$ to ensure a consistent comparison across different parameter settings. For population size N_l, increasing population size improves population diversity and leads to higher HV values, but it also substantially increases computational time. When N_l increases from 50 to 200, the run time increases from 10.56 to 56.25 min, while the HV improvement gradually becomes marginal, indicating limited additional benefit from further enlarging the population. For the maximum number of generations $G_l^{\max}$, after $G_l^{\max} = 100$, the HV curves show only limited room for further improvement, while the runtime continues to increase noticeably. This indicates that the algorithm can achieve sufficient convergence when $G_l^{\max} = 100$. For the crossover probability p_c^l, a relatively high crossover probability helps to accelerate convergence in the early stages. However, an excessively high probability of crossover may cause frequent recombination of individual structures and weaken search stability. The experimental results show that $p_c^l = 0.8$ already achieves a good convergence performance, whereas increasing it to $p_c^l = 1.0$ does not bring about an obvious improvement. For mutation probability p_m^l, an excessively low mutation probability can result in insufficient population diversity and limit the final HV improvement. In contrast, the differences in HV among $p_m^l = 0.1$, 0.2, and 0.3 are relatively small. Therefore, the smaller value $p_m^l = 0.1$ is adopted to balance the maintenance of diversity and avoid excessive random perturbation. For the repair limit $n_{\max}$, increasing $n_{\max}$ from 1 to 10 significantly improves HV, whereas further increasing it to 15 or 20 provides only limited improvement, indicating that the repair capability is nearly saturated around $n_{\max} = 10$. Taking into account the quality of the set of solutions, convergence stability, and computational efficiency, the final parameter settings are determined as $N_l = 50$, $G_l^{\max} = 100$, $p_c^l = 0.8$, $p_m^l = 0.1$, and $n_{\max} = 10$.

Table 8. Average computational time under different lower-level parameter settings.

N_l	Time	$G_l^{\max}$	Time	$n_{\max}$	Time	p_c^l	Time	p_m^l	Time
50	10.56	50	8.54	1	10.56	0.4	10.68	0.05	13.67
100	29.16	100	10.56	5	12.86	0.6	12.02	0.10	14.32
150	43.34	150	20.18	10	14.87	0.8	13.04	0.20	14.68
200	56.25	200	24.51	15	14.84	1.0	14.17	0.30	15.22
–	–	–	–	20	14.50	–	–	–	–

Note: Time is reported in minutes. Each value is averaged over three independent runs.

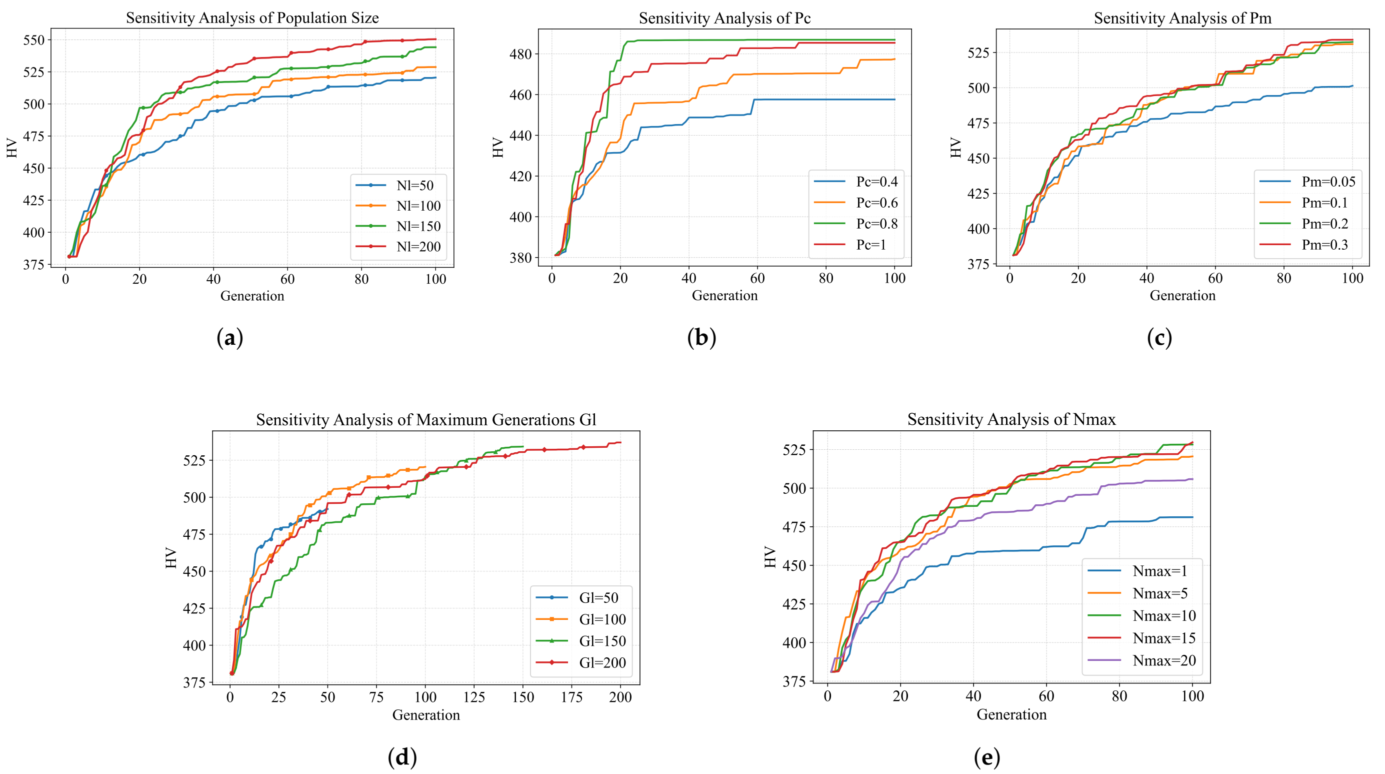

Figure 9. HV convergence curves under different settings of key lower-level parameters: (**a**) population size N_l; (**b**) crossover probability p_c^l; (**c**) mutation probability p_m^l; (**d**) maximum generations $G_l^{\max}$; (**e**) repair limit $n_{\max}$.

5.5.2. Sensitivity Analysis of the Preference Coefficient

To investigate the effect of the task preference coefficient on representative-solution selection, we also performed a sensitivity analysis on ϑ using the S2 scenario. To eliminate the influence of lower-level search fluctuations, the experiment uses the same lower-level non-dominated solution set and applies three values, $\vartheta = 0.2$, $\vartheta = 0.5$, and $\vartheta = 0.8$, for representative-solution selection. The representative solution is then used as input to the upper-level single-UAV sequence refinement stage to produce the final solution. The results are summarized in Table 9.

The results show that ϑ effectively governs the objective preference of the selected solution. When $\vartheta = 0.2$, the algorithm tends to favor solutions with better service-oriented performance; when $\vartheta = 0.8$, it gives greater priority to cost control; and when $\vartheta = 0.5$, the resulting solution exhibits a comparatively balanced trade-off between the two objectives. Because all experiments are conducted on the same lower-level non-dominated solution set, these differences primarily reflect the effect of decision preference on solution selection and on the subsequent local refinement outcome. This confirms that the proposed method is capable of producing decision solutions with different objective emphases from the same non-dominated solution set, depending on the specified preference.

Table 9. Representative and final solutions under different preference coefficients.

ϑ	Representative Solution	Final Solution
0.5	[257.285, 75.8%]	[256.837, 79.0%]
0.2	[260.074, 82.2%]	[259.276, 87.2%]
0.8	[255.965, 65.0%]	[254.595, 61.2%]

5.5.3. Sensitivity Analysis of Resource Availability

To further address the practical applicability of TeCoR-UAV in real-world UAV logistics, we perform an additional sensitivity analysis on the availability of fleet resources. In practical delivery operations, the available UAV fleet may be more abundant than the baseline fleet configuration used for a specific task instance, and an effective planning method should be able to selectively use additional resources rather than blindly activating all available UAVs. Therefore, the original fleet configurations in Table 3 are considered the resource-constrained condition, while the resource-abundant condition is constructed by doubling the number of each type of UAV in the corresponding scenario. The task nodes, obstacle environments, logistics demands, and time-window constraints remain unchanged.

Table 10 reports the results under the two resource-availability conditions. The resource abundance condition generally improves the quality of the Pareto-set, especially in medium- and large-scale scenarios, indicating that additional UAV resources enlarge the feasible search space and provide more flexible allocation choices. Meanwhile, representative solutions do not simply pursue the use of more UAVs to maximize service quality; instead, they adjust the cost–service trade-off according to the available fleet configuration and often form lower-cost delivery plans under abundant resources. In small-scale scenarios, the results under the two resource conditions are close, suggesting that redundant UAVs bring limited additional benefit when the original fleet is already sufficient. Overall, this experiment shows that TeCoR-UAV can adapt to different fleet-resource conditions and effectively exploit abundant resources for better overall optimization, rather than blindly wasting redundant UAV resources.

Table 10. Results of resource abundance sensitivity analysis.

Scenario	Indicator	Resource-Abundant Condition	Resource-Constrained Condition
S1	Average	[80.093, 89.9%]	[80.093, 89.9%]
	Average HV	2.473	2.518
S2	Average	[259.690, 78.7%]	[220.040, 75.3%]
	Average HV	9.913	1.796

Table 10. *Cont.*

Scenario	Indicator	Resource-Abundant Condition	Resource-Constrained Condition
S3	Average Average HV	[383.300, 72.6%] 15.293	[423.236, 75.6%] 5.451
S4	Average Average HV	[81.350, 99.9%] 19.583	[84.074, 100.0%] 19.923
S5	Average Average HV	[216.514, 77.2%] 9.743	[256.900, 87.0%] 4.206
S6	Average Average HV	[368.401, 76.1%] 18.810	[426.915, 84.4%] 5.592

6. Conclusions

This paper proposes TeCoR-UAV, a cooperative delivery optimization method for complex low-altitude logistics environments with multiple UAVs. The method aims to connect airspace accessibility representation to multi-UAV cooperative route planning, so that the resulting plans satisfy multiple complex constraints while more faithfully reflecting flight costs in complex airspace. Therefore, TeCoR-UAV provides a cooperative planning solution for low-altitude logistics that jointly considers feasibility, service quality, and operational cost.

The experiments were conducted under scenarios with different scales and topological conditions. The results show that TeCoR-UAV achieves a more stable overall optimization performance and higher Pareto-set quality in medium- and large-scale scenarios, as well as in topologically constrained scenarios. In comparison, conventional multi-objective optimization methods are limited in feasible-solution search when complex constraints are strongly coupled. Learning-based methods can obtain high-quality feasible solutions, but their solving efficiency decreases noticeably in large-scale complex scenarios. Compared with the online path-planning method D-LOF, TeCoR-UAV reduces repeated obstacle-avoidance path searches through offline topology construction and path-cost reuse, showing better computational affordability. Further ablation and sensitivity analyses verify the necessity of the key components of TeCoR-UAV and the overall method's scenario adaptability. These results indicate that the proposed method can improve search stability and Pareto-set quality, while flexibly adjusting the planning results under different fleet-resource conditions.

Although TeCoR-UAV has shown promising optimization performance, several aspects can still be extended and improved. Future work will mainly focus on three directions: (1) further investigating online topology updating and real-time replanning in dynamic low-altitude environments to adapt to temporary no-fly zones, task changes, and environmental variations; (2) further improving flight-cost modeling by incorporating factors such as wind fields and communication constraints into the planning process, thus improving the representation of real low-altitude logistics scenarios; and (3) conducting real-flight experiments to further validate the engineering applicability, robustness, and practical deployment capability of TeCoR-UAV.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics15132939/s1, Dataset S1: Datasets corresponding to the experimental scenarios used in the simulation experiments.

Author Contributions: B.D. and W.N. designed the core algorithms and wrote the main manuscript. M.Z. conceived the overall architecture and supervised the study. Y.L., N.L. and J.L. provided methodological guidance and revised the manuscript. Z.L. provided supervision and resources. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The source code, benchmark instances, parameter settings, and scripts used to reproduce the experimental results are publicly available in the project repository at https://github.com/byding823-cmd/TeCoR-UAV.git, (accessed on 29 June 2026).

Conflicts of Interest: The authors declare that they have no competing interests related to the content of this article.

Appendix A

Table A1. Main parameter settings used in the Stage I topology extraction module.

Parameter	Value
Path-length weight ϖ_1 for $\Lambda_{ij}^{\mathrm{len}}$	1
Turning-angle smoothness weight ϖ_2 for $\Upsilon_{ij}^{\mathrm{turn}}$	1
Constraint-violation penalty weight ϖ_3 for Ξ_{ij}^{viol}	1
Objective function	$\min_{\pi_{ij}} \mathfrak{J}_{ij}^{\mathrm{PSO}} = \varpi_1 \Lambda_{ij}^{\mathrm{len}} + \varpi_2 \Upsilon_{ij}^{\mathrm{turn}} + \varpi_3 \Xi_{ij}^{\mathrm{viol}}$
Number of particles	50
Maximum iterations	40
Cognitive acceleration coefficient	1.8
Social acceleration coefficient	1.8
Inertia weight	Linearly decreased from 0.9 to 0.4
Number of intermediate waypoints	6
Initial waypoint perturbation in x and y	0.3
Initial waypoint perturbation in z	0.05
B-spline degree	3
Number of smoothing samples	100
Trajectory length evaluation	Segment-length summation over the sampled smoothed trajectory

References

1. Garg, V.; Niranjan, S.; Prybutok, V.; Pohlen, T.; Gligor, D. Drones in Last-Mile Delivery: A Systematic Review on Efficiency, Accessibility, and Sustainability. *Transp. Res. Part D Transp. Environ.* **2023**, *123*, 103831. [CrossRef]
2. Huang, C.; Fang, S.; Wu, H.; Wang, Y.; Yang, Y. Low-Altitude Intelligent Transportation: System Architecture, Infrastructure, and Key Technologies. *J. Ind. Inf. Integr.* **2024**, *42*, 100694. [CrossRef]
3. Rejeb, A.; Rejeb, K.; Simske, S.J.; Treiblmaier, H. Drones for Supply Chain Management and Logistics: A Review and Research Agenda. *Int. J. Logist. Res. Appl.* **2021**, *26*, 708–731. [CrossRef]
4. Raghunatha, A.; Lindkvist, E.; Thollander, P.; Hansson, E.; Jonsson, G. Critical Assessment of Emissions, Costs, and Time for Last-Mile Goods Delivery by Drones Versus Trucks. *Sci. Rep.* **2023**, *13*, 11814. [CrossRef] [PubMed]
5. Sharma, S.; Sharma, H. Drone a Technological Leap in Health Care Delivery. *J. Fam. Med. Prim. Care* **2024**, *13*, 302–308. [CrossRef] [PubMed]
6. Hu, Z.H.; Li, T.; Tian, X.D.; Wei, Y.H. Drone-Based Emergent Distribution of Packages to an Island from a Land Base. *Drones* **2023**, *7*, 218. [CrossRef]
7. Dukkanci, O.; Koberstein, A.; Kara, B.Y. Drones for Relief Logistics Under Uncertainty After an Earthquake. *Eur. J. Oper. Res.* **2023**, *310*, 117–132. [CrossRef]
8. Borghetti, F.; Caballini, C.; Carboni, A.; Grossato, G.; Maja, R.; Barabino, B. The Use of Drones for Last-Mile Delivery: A Numerical Case Study in Milan, Italy. *Sustainability* **2022**, *14*, 1766. [CrossRef]
9. Yang, Y.; Fu, Y.; Lu, D.; Xiang, H.; Xu, K. Three-Dimensional Unmanned Aerial Vehicle Trajectory Planning Based on the Improved Whale Optimization Algorithm. *Symmetry* **2024**, *16*, 1561. [CrossRef]

10. Du, P.; He, X.; Cao, H.; Garg, S.; Kaddoum, G.; Hassan, M.M. AI-Based Energy-Efficient Path Planning of Multiple Logistics UAVs in Intelligent Transportation Systems. *Comput. Commun.* **2023**, *207*, 46–55.
11. Wang, Y.; Yang, S.; Wang, X.V.; Wang, L. Research on Truck-Drone Collaborative Route Planning for Rural Logistics Delivery Services. *Sci. Rep.* **2024**, *14*, 31815. [CrossRef] [PubMed]
12. Liu, B.; Lyu, Y. A 3D UAV Path Planning Model Based on Improved A* Algorithm and DEM Data. *J. Physics Conf. Ser.* **2023**, *2580*, 012043. [CrossRef]
13. Lin, Y.; Saripalli, S. Sampling-Based Path Planning for UAV Collision Avoidance. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3179–3192. [CrossRef]
14. Qi, Y.; Jiang, H.; Huang, G.; Yang, L.; Wang, F.; Xu, Y. Multi-UAV Path Planning Considering Multiple Energy Consumptions via an Improved Bee Foraging Learning Particle Swarm Optimization Algorithm. *Sci. Rep.* **2025**, *15*, 14755. [CrossRef] [PubMed]
15. Dorling, K.; Heinrichs, J.; Messier, G.G.; Magierowski, S. Vehicle Routing Problems for Drone Delivery. *IEEE Trans. Syst. Man, Cybern. Syst.* **2017**, *47*, 70–85. [CrossRef]
16. Zhang, H.; Dou, L.; Xin, B.; Zhang, R.; Wang, Q. Reconnaissance and Confirmation Task Planning of Multiple Fixed-Wing UAVs with Specific Payloads: A Comparison Study. *J. Adv. Comput. Intell. Intell. Informatics* **2022**, *26*, 570–580. [CrossRef]
17. Hong, Y.; Kim, S.; Kim, Y.; Cha, J. Quadrotor Path Planning Using A* Search Algorithm and Minimum Snap Trajectory Generation. *ETRI J.* **2021**, *43*, 1013–1023. [CrossRef]
18. Han, Z.; Chen, M.; Shao, S.; Wu, Q. Improved Artificial Bee Colony Algorithm-Based Path Planning of Unmanned Autonomous Helicopter Using Multi-Strategy Evolutionary Learning. *Aerosp. Sci. Technol.* **2022**, *122*, 107374. [CrossRef]
19. Yu, X.; Jiang, N.; Wang, X.; Li, M. A Hybrid Algorithm Based on Grey Wolf Optimizer and Differential Evolution for UAV Path Planning. *Expert Syst. Appl.* **2023**, *215*, 119327. [CrossRef]
20. Huang, S.; Tian, J.; Qiao, L.; Wang, Q.; Su, Y. Unmanned Aerial Vehicle Path Planning Based on Improved Genetic Algorithm. *J. Comput. Appl.* **2021**, *41*, 390–397. (In Chinese) [CrossRef]
21. Ait-Saadi, A.; Meraihi, Y.; Soukane, A.; Ramdane-Cherif, A.; Benmessaoud Gabis, A. A Novel Hybrid Chaotic Aquila Optimization Algorithm with Simulated Annealing for Unmanned Aerial Vehicles Path Planning. *Comput. Electr. Eng.* **2022**, *104*, 108461. [CrossRef]
22. Katada, Y.; Kato, T.; Nakajima, N. Route Planning of an Unmanned Aerial Vehicle Using Ant Colony Optimization for Environment Including a No-Fly Zone and Obstacles. *Artif. Life Robot.* **2026**, *31*, 254–264. [CrossRef]
23. Guo, Y.; Liu, X.; Jia, Q.; Liu, X.; Zhang, W. HPO-RRT*: A Sampling-Based Algorithm for UAV Real-Time Path Planning in a Dynamic Environment. *Complex Intell. Syst.* **2023**, *9*, 7133–7153. [CrossRef]
24. Xu, X.; Zhang, F.; Zhao, Y. Unmanned Aerial Vehicle Path-Planning Method Based on Improved P-RRT* Algorithm. *Electronics* **2023**, *12*, 4576. [CrossRef]
25. ElSayed, M.; Mohamed, M. Robust Digital-Twin Airspace Discretization and Trajectory Optimization for Autonomous Unmanned Aerial Vehicles. *Sci. Rep.* **2024**, *14*, 12506. [CrossRef] [PubMed]
26. Turker, T.; Yilmaz, G.; Sahingoz, O.K. GPU-Accelerated Flight Route Planning for Multi-UAV Systems Using Simulated Annealing. In *Artificial Intelligence: Methodology, Systems, and Applications*; Springer International Publishing: Cham, Switzerland, 2016; pp. 279–288.
27. Kuo, R.J.; Edbert, E.; Zulvia, F.E.; Lu, S.H. Applying NSGA-II to Vehicle Routing Problem with Drones Considering Makespan and Carbon Emission. *Expert Syst. Appl.* **2023**, *221*, 119777. [CrossRef]
28. Yu, V.F.; Anh, P.T.; Gunawan, A.; Han, H. A Simulated Annealing with Variable Neighborhood Descent Approach for the Heterogeneous Fleet Vehicle Routing Problem with Multiple Forward/Reverse Cross-Docks. *Expert Syst. Appl.* **2024**, *237*, 121631. [CrossRef]
29. Song, B.D.; Park, K.; Kim, J. Persistent UAV Delivery Logistics: MILP Formulation and Efficient Heuristic. *Comput. Ind. Eng.* **2018**, *120*, 418–428. [CrossRef]
30. Amirteimoori, A.; Kia, R.; Mohamed, M.; Weber, G.W. An Innovative Framework Integrating MILP and a Parallel Optimal Algorithm for UAV-Enabled Last-Mile Delivery. *Int. J. Prod. Res.* **2025**, *64* , 777–797. [CrossRef]
31. Puente-Castro, A.; Rivero, D.; Pazos, A.; Fernandez-Blanco, E. Using Reinforcement Learning in the Path Planning of Swarms of UAVs for the Photographic Capture of Terrains. *Eng. Proc.* **2021**, *7*, 32. [CrossRef]
32. Ghaffar, M.A.; Peng, L.; Aslam, M.U.; Adeel, M.; Dassari, S. Vehicle-UAV Integrated Routing Optimization Problem for Emergency Delivery of Medical Supplies. *Electronics* **2024**, *13*, 3650. [CrossRef]
33. Cicek, D.; Simsek, M.; Kantarci, B. Machine Learning-Driven Truck–Drone Collaborative Delivery for Time- and Energy-Efficient Last-Mile Deliveries. *Electronics* **2025**, *14*, 2026. [CrossRef]
34. Fang, Z.; Ma, T.; Huang, J.; Niu, Z.; Yang, F. Efficient Task Allocation in Multi-Agent Systems Using Reinforcement Learning and Genetic Algorithm. *Appl. Sci.* **2025**, *15*, 1905. [CrossRef]
35. Liu, Y.; Shi, J.; Luo, Z.; Hu, X.; Pedrycz, W.; Liu, Z. Cooperated Truck-Drone Routing With Drone Energy Consumption and Time Windows. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 20390–20404. [CrossRef]

36. Cong, Y.; Bie, Y.; Liu, Z.; Zhu, A. Collaborative Vehicle-Crew Scheduling for Multiple Routes with a Mixed Fleet of Electric and Fuel Buses. *Energy* **2024**, *298*, 131400. [CrossRef]
37. Zhou, B.; Zeng, W.; Yang, H. Multi-Trip UAV-UGV Delivery Network Design with Release Times. *Transp. Res. Part C: Emerg. Technol.* **2025**, *181*, 105389. [CrossRef]
38. Hong, Y.; Wu, Q. Dual-Stage Collaborative Path Planning and Task Allocation for UAV Swarms in Complex Maritime Rescue Environments. *Reliab. Eng. Syst. Saf.* **2026**, *268*, 111974. [CrossRef]
39. Li, J.; Cang, L.; Wu, Y.; Zhang, Z. Two-Echelon Collaborative Many-to-Many Pickup and Delivery Problem for Agricultural Wholesale Markets with Workload Balance. *Omega* **2025**, *130*, 103164. [CrossRef]
40. Wang, Y.; Leixin, J.; Zhang, D.; Ji, Y.; Wang, X.V.; Wang, L. A Dual-Layer Optimization Framework for Multi-UAV Delivery Scheduling in Multi-Altitude Urban Airspace. *Drones* **2026**, *10*, 203. [CrossRef]
41. Torabbeigi, M.; Lim, G.J.; Kim, S.J. Drone Delivery Scheduling Optimization Considering Payload-Induced Battery Consumption Rates. *J. Intell. Robot. Syst.* **2020**, *97*, 471–487. [CrossRef]
42. ACSL Ltd. *AirTruck: Dedicated Logistics Drone*; ACSL Ltd.: Tokyo, Japan, 2026.
43. Wingcopter GmbH. *Wingcopter 198*; Wingcopter GmbH: Weiterstadt, Germany, 2026.
44. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]