

Article

Not peer-reviewed version

A Bi-Layer Collaborative Planning Framework for Multi-UAV Delivery Tasks in Multi-Depot Urban Logistics

Junfu Wen , [Fei Wang](#) ^{*} , Yebo Su

Posted Date: 12 June 2025

doi: 10.20944/preprints202506.1002.v1

Keywords: Urban logistics distribution; Multi-UAV operation; Multi-depot coordination; Bi-layer collaborative optimization; Enhanced NSGA-II algorithm



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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

Article

A Bi-Layer Collaborative Planning Framework for Multi-UAV Delivery Tasks in Multi-Depot Urban Logistics

Junfu Wen ^{†,‡} , Fei Wang ^{*,‡} and Yebo Su

College of Air Traffic Management, Civil Aviation University of China, Tianjin 300300, China

* Correspondence: fei.wang@cauc.edu.cn

[†] Current address: 2898 Jinbei Road · Dongli District.

[‡] These authors contributed equally to this work.

Abstract: To address the modeling complexity and multi-objective collaborative optimization challenges in multi-depot and multiple Unmanned Aerial Vehicle (UAV) delivery task planning, this paper proposes a bi-layer planning framework, which comprehensively considers resource constraints, multi-depot coordination, and the coupling characteristics of path execution. The upper layer tackles the task assignment problem by formulating a multi-objective optimization model aimed at minimizing economic cost, delivery delay, and the number of UAVs deployed. An Enhanced Non-dominated Sorting Genetic Algorithm II (ENSGA-II) is developed, incorporating heuristic initialization, goal-oriented search operators, an adaptive mutation mechanism, and a staged evolution control strategy to improve solution feasibility and distribution quality. The lower layer addresses the path planning problem by establishing a multi-objective model that considers path length, flight risk, and altitude variation. An improved Particle Swarm Optimization (PSO) algorithm is proposed by integrating chaotic initialization, linearly adjusted acceleration coefficients and maximum velocity, a stochastic disturbance-based position update mechanism, and an adaptively tuned inertia weight to enhance algorithmic performance and path generation quality. Simulation results under typical task scenarios demonstrate that the proposed model achieves an average reduction of 47.8% in economic cost and 71.4% in UAV deployment quantity, while significantly reducing delivery window violations. The framework exhibits excellent capability in multi-objective collaborative optimization. The ENSGA-II algorithm outperforms baseline algorithms significantly across performance metrics, achieving a hypervolume (HV) value of 1.0771 (improving by 72.35% to 109.82%) and an average Inverted Generational Distance (IGD) of 0.0295, markedly better than those of comparison algorithms (ranging from 0.0893 to 0.2714). The algorithm also demonstrates overwhelming superiority in the C-metric, indicating outstanding global optimization capability in terms of distribution, convergence, and diversity of the solution set. Moreover, the proposed framework and algorithm are both effective and feasible, offering a novel approach to low-altitude urban logistics delivery problems.

Keywords: urban logistics distribution; multi-UAV operation; multi-depot coordination; Bi-layer collaborative optimization; enhanced NSGA-II algorithm

1. Introduction

With continuous advancements in intelligent sensing, navigation, and communication technologies, multiple Unmanned Aerial Vehicles (UAV) systems have demonstrated significant application potential in fields such as urban delivery, emergency response, and inspection operations [1]. In particular, multi-UAV logistics distribution has emerged as a key research focus within intelligent transportation systems, attracting considerable attention from both academia and industry due to its rapid, efficient, and flexible cargo transportation capabilities. Fundamentally, the multi-UAV logistics distribution problem represents a complex combinatorial optimization challenge that involves several tightly coupled subproblems, such as task assignment, route planning, and resource scheduling.

These problems are typically characterized by high dimensionality, strong nonlinearity, and multiple constraints [2].

The multi-UAV logistics distribution problem is widely regarded as a representative application of multi-UAV task planning in practical scenarios. Both logistics distribution and task planning share high structural and decision-making consistency, encompassing essential elements such as task allocation, route generation, and coordinated scheduling. From a modeling perspective, these problems require the integration of spatial topology, time window constraints, task priorities, and flight resource limitations into formalized models that reflect the operational characteristics of multi-UAV systems. At the algorithmic layer, task allocation and route planning are inherently interdependent and must be jointly optimized to enhance system-wide performance. Accordingly, the modeling and solution paradigms for multi-UAV logistics distribution exhibit substantial convergence with those for general task planning [3]. For instance, Wang et al. [6] proposed a resilient planning framework that incorporates task pre-assignment and reallocation modules, and adopted an improved genetic algorithm for efficient optimization. Chang et al. [7] developed a cooperative search framework for UAV inspection route planning, combining regional grid-based decentralization and an adaptive initial solution generation strategy. Ahmed et al. [8] introduced a metaheuristic-based energy-efficient path planning method for UAV networks, aimed at minimizing energy consumption and optimizing coordination in densely obstructed environments. Li et al. [9] investigated the impact of wind fields in three-dimensional mountainous environments, constructing a wind-aware sensing space and designing a multi-objective cost function, along with an enhanced swarm intelligence algorithm. Although these methods show good effectiveness in small-scale problems, their performance deteriorates as task scale and UAV heterogeneity increase. Centralized approaches often fail to respond effectively to dynamic task environments and system fluctuations [10]. Moreover, such approaches lack structured information decomposition mechanisms, limiting their capacity to coordinate task-vehicle matching with path execution. Consequently, they exhibit limited adaptability and scalability in complex, large-scale, and constraint-intensive scenarios [11].

To address the aforementioned challenges, the development of hierarchical optimization models with explicit separation between the task layer and the routing layer has become a prominent research direction in multi-UAV task planning problems [12]. Specifically, the upper layer is responsible for determining task assignment strategies under global resource constraints, while the lower layer focuses on generating feasible routes for individual UAVs. These two layers not only pursue different optimization objectives but are also interconnected through bidirectional information flow and feedback mechanisms. Structuring such problems within a bi-layer optimization framework that explicitly delineates functional responsibilities facilitates the reduction of modeling complexity and solution dimensionality. Moreover, it enhances the adaptability of the system to dynamic task changes and heterogeneous resources, and improves the interpretability and practical applicability of scheduling strategies [13]. For instance, Liu et al. [14] proposed a bi-layer adaptive coordination algorithm based on individual and group decision-making, aiming to satisfy the requirements of trajectory planning and formation flight. Cheng et al. [15] developed a distributed path planning approach for multi-UAV systems based on a bi-layer coordination framework, addressing limitations of conventional methods such as low computational efficiency, poor scalability, and difficulties in collision avoidance under obstacle-rich environments. Yan [16] introduced an enhanced multi-type gene genetic algorithm to efficiently resolve the coupled problem of task assignment and route planning in cooperative UAV attack missions. Zhan et al. [17] proposed a genetic algorithm integrated with reinforcement learning (GA-RL) to jointly optimize task allocation and path planning in maritime multi-UAV search and rescue missions. Their approach employed dynamic population management and adaptive strategies to improve the rationality of task distribution and the efficiency of path search. However, most existing studies are based on a single-depot configuration and fail to address the practical demand for coordinated scheduling among multiple depots.

In urban logistics, emergency response, and distributed operation scenarios, multiple depots typically participate simultaneously in task distribution and resource coordination. This leads to more complex spatial structures and competitive task environments. Under such conditions, traditional single-depot bi-layer models encounter significant limitations in adaptability. On one hand, dynamic interactions such as resource sharing, task reassignment, and cross-regional scheduling among depots introduce coupling characteristics that cannot be effectively captured by simple task partitioning and independent route planning. On the other hand, routing problems under multi-depot settings often involve challenges such as cross-region task coverage, route synchronization, and service priority balancing, which impose more demanding requirements on both modeling capacity and computational efficiency [18]. Furthermore, the presence of multiple depots introduces more intricate real-world constraints. These include heterogeneous resource capabilities and coverage radii across depots, significantly uneven task point distributions, and the need to accommodate varying UAV flight performance, payload capacities, and configuration requirements under multi-task and heterogeneous matching mechanisms. These compounded factors intensify the dependency between the task and routing layers, further exacerbating performance degradation in decoupled modeling approaches [19]. Therefore, it is imperative to construct a bi-layer optimization framework tailored to the multi-depot multi-UAV scenario. Such a framework should comprehensively capture multi-depot resource coordination mechanisms at the task layer, systematically integrate UAV heterogeneity and operational constraints at the routing layer, and enhance the feedback mechanisms between the two layers. This will facilitate the development of more intelligent, efficient, and scalable scheduling strategies.

As mentioned above, this study addresses the modeling complexity and multi-objective coordination challenges inherent in the bi-layer task planning problem for multi-Depot multi-UAV systems by developing a hierarchical task planning model. The upper-layer model focuses on task allocation, involving coordination among multiple depots and UAVs. The objective is to minimize the total economic cost, cumulative task delay time, and the number of UAVs required. The model incorporates constraints including the maximum allowable delay time, depot closure time windows, UAV capacity and fleet size limitations, and the maximum flight path length. The lower-layer model is designed for route planning, taking into account multiple factors such as total path length, exposure risk in hazardous grid regions, and the cost of altitude variation. Additional constraints, including altitude interval restrictions and climb rate limits, are introduced to improve the feasibility and safety of the planned routes. To solve the proposed model, an Enhanced Non-dominated Sorting Genetic Algorithm II (ENSGA-II) is developed. This algorithm integrates a heuristic initialization method based on route feasibility constraints, a bi-directional search operator framework guided by optimization objectives, and an adaptive mutation strategy. This design enables a balanced integration of global exploration and local exploitation, enhancing convergence quality, solution diversity, and the overall robustness and efficiency of the multi-objective optimization process. This study provides a novel and systematic modeling and optimization framework for complex scheduling problems in multi-UAV multi-depot scenarios, offering both significant theoretical contributions and practical application potential.

The remainder of this paper is organized as follows. Section 2 introduces the modeling of the urban spatial environment for delivery tasks, establishing the spatial structure and constraint conditions required for task planning. Section 3 presents a formal model of the bi-layer task planning problem, including detailed formulations of the objective functions and constraints for both the task assignment and route planning layers, as well as the design principles and key components of the proposed ENSGA-II algorithm. Section 4 conducts simulation experiments and performance evaluations from multiple perspectives. Section 5 summarizes the research findings and discusses potential directions for future work.

2. Urban Environment Modeling

In multi-UAV systems designed for urban-scale delivery operations, constructing a rational and structured environmental model is essential for subsequent task allocation, path planning, and

safety evaluation. Due to the spatial complexity of low-altitude urban airspace, characterized by dense building layouts and static obstacles, direct modeling and abstraction of the three-dimensional (3D) space can significantly enhance the computational tractability and algorithmic feasibility of the planning process. The core objective of environment modeling is to structurally convert the actual physical airspace into a computationally processable spatial representation, thereby enabling a mathematical formulation of complex environmental information. Existing urban environment modeling approaches predominantly include grid-based discretization, graph-theoretical modeling, and visibility graph construction. Among them, the grid-based method has gained wide adoption in low-altitude UAV flight scenarios due to its simplicity, flexibility, and ability to support diverse spatial attribute definitions.

In this study, a three-dimensional grid-based discretization method [28] is employed to model the urban delivery task area. Let the operational airspace be a cuboidal region with side lengths x , y , and z , respectively. This space is discretized into uniform cubic grid cells of side length l_{grid} . The total number of grids in each dimension is calculated as follows:

$$m = \left\lfloor \frac{x}{l_{\text{grid}}} \right\rfloor, \quad n = \left\lfloor \frac{y}{l_{\text{grid}}} \right\rfloor, \quad h = \left\lfloor \frac{z}{l_{\text{grid}}} \right\rfloor \quad (1)$$

where $\lfloor \cdot \rfloor$ denotes the floor operation. The resulting 3D grid structure forms a discretized spatial set of dimension $m \times n \times h$. Figure 1 illustrates a schematic of the space partitioning process. Each grid cell is indexed by the coordinates of its centroid and can be assigned multi-dimensional attributes such as obstacle presence, path cost, and risk layer. The selection of grid resolution l_{grid} directly influences the trade-off between path planning accuracy and computational cost. A coarse grid resolution may result in insufficient path granularity or missed narrow passageways, whereas an overly fine resolution substantially increases computational burden, potentially compromising real-time performance and scalability.

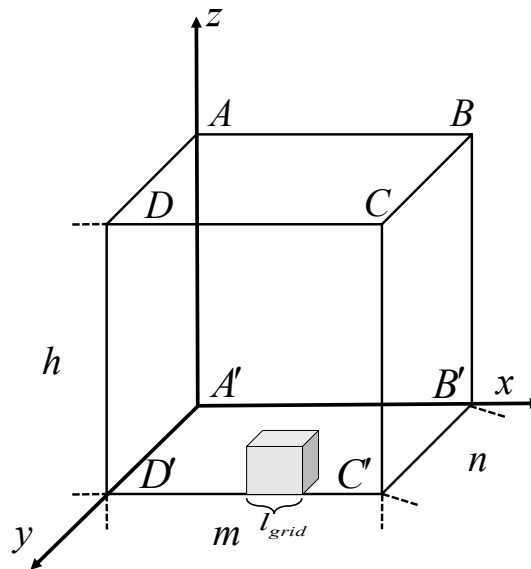


Figure 1. 3-D environment modeling by grid-based method.

In addition, common obstacles in low-altitude urban operational environments include high-rise buildings, telecommunication towers, elevated bridges, and dense vegetation. To streamline the modeling process, all obstacles in this study are uniformly abstracted as cuboid voxel blocks with defined spatial boundaries, and mapped to the corresponding positions within the grid-based environmental representation. Traditional modeling approaches generally adopt a binary labeling method, wherein grid cells corresponding to obstacles are assigned a value of 1, indicating impassable

regions, while remaining cells are assigned a value of 0, indicating traversable free space. Although this method offers implementation simplicity, it suffers from a significant limitation, as it does not provide a quantitative assessment of the risk associated with proximity to obstacle regions, thereby restricting its capacity for supporting dynamic safety evaluations.

To overcome this limitation, this study introduces a risk-based grid model derived from neighborhood relations [29], which quantitatively assesses the flight safety of traversable grid cells. Specifically, for each non-obstacle grid cell denoted as k , the associated risk value is defined as the proportion of its adjacent neighboring cells that are occupied by obstacles. This relationship is formally expressed as follows:

$$O_k = \frac{N_{obs}^k}{N_{total}^k} \quad (2)$$

where N_{obs}^k denotes the number of neighboring cells occupied by obstacles, and N_{total}^k indicates the total number of neighboring cells surrounding grid cell k . This risk index effectively captures the local hazard layer associated with traversable regions, and is subsequently integrated into the objective function of the path planning model to achieve a balance between navigation efficiency and environmental risk avoidance. Figure 2 presents the spatial distribution of grid-based risk values. Grey-shaded areas denote obstacle cells with a risk value of 1, while white cells represent non-obstacle regions whose risk layers dynamically adjust based on the density of adjacent obstacle cells. This approach preserves the discrete modeling advantages of conventional grid-based methods while substantially enhancing the granularity with which potential environmental hazards can be represented [30]. In conclusion, the proposed risk-informed grid-based environmental modeling approach establishes a rigorous and extensible spatial data foundation for the subsequent multi-agent scheduling and trajectory optimization models. It contributes significantly to the operational robustness and navigational safety of urban airspace delivery systems.

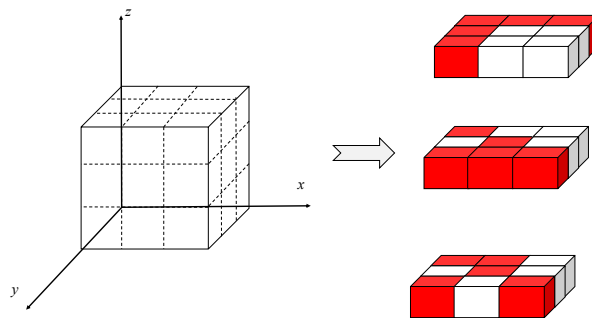


Figure 2. Schematic illustration of risk distribution in the environmental model.

3. Bi-Layer Collaborative Task Planning Model and Algorithm

This section establishes a bi-layer collaborative task planning framework for multi-UAV and multi-distribution center scenarios, as illustrated in Figure 3. The upper-layer model is responsible for global task allocation and resource configuration. It comprehensively considers task urgency, UAV performance, and distribution center locations to optimally match tasks, UAV resources, and delivery routes. The lower-layer model focuses on path planning and optimization for each UAV that has been assigned tasks. Its objective is to minimize flight cost and delay penalties, thereby improving delivery efficiency and resource utilization. A collaborative optimization mechanism is achieved through information exchange between the two layers: the task allocation results from the upper layer serve as inputs for the lower layer, while the lower layer feeds back key indicators such as flight distance to assist the upper layer in iterative optimization. This forms a closed-loop optimization process. The following subsections provide a detailed description of the construction methods and optimization strategies for both layers of the model.

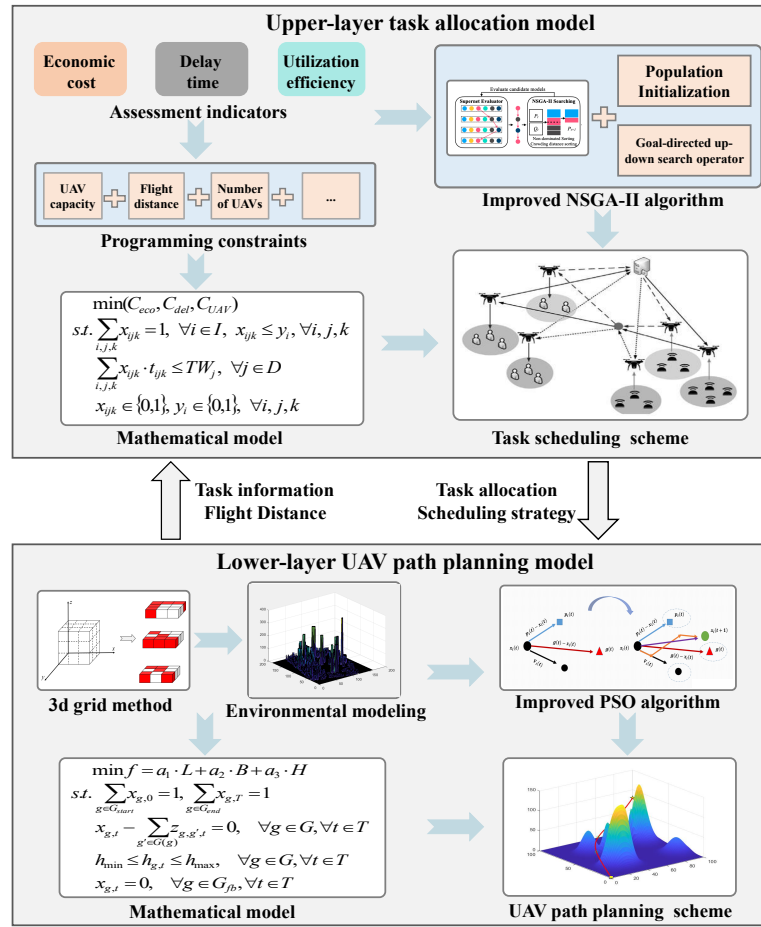


Figure 3. The framework of the bi-layer coordinated task planning model.

3.1. Upper-Layer Task Allocation Model and Algorithm

Figure 4 illustrates the structural schematic of the upper-layer task allocation model. Within the bi-layer collaborative task planning framework proposed in this study, the upper-layer model focuses on global scheduling decisions involving multiple distribution centers, multiple task points, and multiple UAVs. This process takes into account several critical factors: the spatially discrete distribution of task points, the deployment locations of distribution centers, the heterogeneous performance parameters of UAVs (such as endurance, maximum payload, flight speed, etc.), as well as overall scheduling costs and timeliness requirements of the system. By considering these aspects, the model aims to achieve multi-objective collaborative optimization, balancing service quality, resource utilization efficiency, and economic cost.

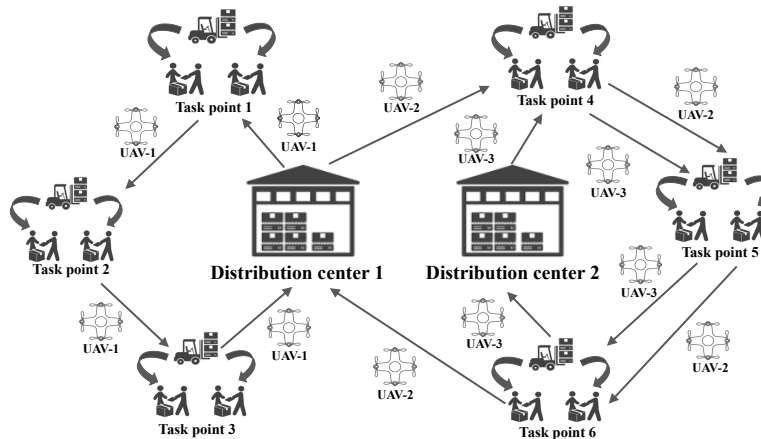


Figure 4. Schematic diagram of multi-depot and multi-UAV task allocation structure.

3.1.1. Mathematical Model

(1) Decision variables

In the upper-layer task assignment model, let the set of all delivery tasks be denoted by I , where $i = 1, 2, \dots, |I|$. The set of distribution centers involved in scheduling is denoted by J , where $j = 1, 2, \dots, |J|$. The available UAV types form the UAV model set K , where $k = 1, 2, \dots, |K|$. To describe the assignment and scheduling status in the multi-depot multi-UAV system, the following key decision variables are defined to capture the matching relationship among tasks, distribution centers, and UAVs:

$$x_{ijk} = \begin{cases} 1, & \text{if task } i \text{ is assigned to UAV type } k \text{ at depot } j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$y_{jk} = \begin{cases} 1, & \text{if depot } j \text{ dispatches a UAV of type } k \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

(2) Objective functions

This section develops a multi-objective optimization model incorporating economic cost, delay time, and UAV resource utilization, aiming to minimize overall operational cost and resource consumption while ensuring timely task completion. Specifically, the objective function comprises the following three sub-objectives:

1) Economic cost:

The economic cost function consists of three components: the scheduling cost induced by task waiting time, the fixed cost incurred by UAV deployment, and the variable cost associated with flight path length, formulated as:

$$\min C_{eco} = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \omega_i \cdot (t_i^s - r_i) \cdot x_{ijk} + \sum_{j \in J} \sum_{k \in K} f_k \cdot y_{jk} + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} v_k \cdot D_{ij} \cdot x_{ijk} \quad (5)$$

where ω_i denotes the unit waiting cost of task i , t_i^s and r_i represent the delivery start time and request time of task i , respectively, f_k and v_k denote the fixed scheduling cost and unit distance cost of UAV type k , respectively, and D_{ij} is the distance from depot j to task point i .

2) Total delay time:

To ensure service timeliness, delivery tasks must be completed within the time windows specified by customers. Each task point i is assigned a service time window $[e_i, d_i]$, where e_i denotes the earliest allowable service time and d_i denotes the latest permissible service time. If the actual arrival time of the UAV at task point i exceeds d_i , a delay is incurred. The delay time is defined as the difference between the actual arrival time and the latest allowable service time:

$$C_{del} = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \max(0, t_i^c - d_i) \cdot x_{ijk} \quad (6)$$

where t_i^c denotes the actual arrival time at task point i .

3) Number of UAVs:

By minimizing the number of deployed UAVs, the system can effectively allocate platform resources while still meeting task requirements:

$$C_{UAV} = \sum_{j \in J} \sum_{k \in K} y_{jk} \quad (7)$$

Combining the above three objectives yields a vector-based multi-objective optimization problem:

$$\min(C_{eco}, C_{del}, C_{UAV}) \quad (8)$$

(3) Constraints

To ensure the feasibility of the model solution and the rationality of the scheduling results, the following constraints are imposed:

1) Task uniqueness constraint:

This constraint guarantees that each task is performed by exactly one UAV from a specific distribution center, thus maintaining task integrity:

$$\sum_{j \in J} \sum_{k \in K} x_{ijk} = 1, \quad \forall i \in I \quad (9)$$

2) Delay time constraint:

This constraint ensures that a delay occurs only if the actual arrival time at task i exceeds the latest allowable service time in its time window:

$$Del_i = \max(0, t_i^c - d_i) \quad (10)$$

where Del_i denotes the delay time for task i .

3) Distribution center time window constraint:

To align the schedule with the operating hours of each distribution center, it is required that all UAVs return to their corresponding centers before the center's latest closing time after completing their assigned tasks:

$$t_i^e \leq T_j^{clo}, \quad \forall (i, j, k), x_{ijk} = 1 \quad (11)$$

where t_i^e denotes the time at which the UAV returns to its distribution center after completing task i , and T_j^{clo} represents the closing time of distribution center j .

4) UAV load constraint:

This constraint ensures that the total weight of goods assigned to a single UAV does not exceed its maximum carrying capacity:

$$\sum_{i \in I} q_i \cdot x_{ijk} \leq Q_k \cdot y_{jk}, \quad \forall j \in J, k \in K \quad (12)$$

where q_i denotes the weight of goods required to be transported for task i , and Q_k represents the maximum payload capacity of a UAV of type k .

5) UAV quantity constraint:

The maximum number of UAVs dispatched is limited by the available fleet size:

$$\sum_{j \in J} y_{jk} \leq N_k, \quad \forall k \in K \quad (13)$$

where N_k denotes the total number of UAVs currently available.

6) Single-task path length constraint:

To ensure flight safety and power sustainability, the flight path for each task must not exceed the maximum allowable distance:

$$\sum_{j \in J} \sum_{k \in K} D_{ij} \cdot x_{ijk} \leq L^{\max}, \quad \forall i \in I \quad (14)$$

where $L^{\max}$ represents the maximum flight distance allowed for a single task.

7) Binary constraint:

$$x_{ijk} \in \{0, 1\}, \quad y_{jk} \in \{0, 1\} \quad (15)$$

As mentioned above, the multi-depot multi-UAV task allocation problem in the upper-layer model is described as follows:

$$\begin{aligned} & \min (C_{\text{eco}}, C_{\text{deli}}, C_{\text{UAV}}) \\ & \text{s.t.} \begin{cases} \sum_{i,j,k} x_{ijk} = 1, & \forall i \in I \\ x_{ijk} \leq y_j, & \forall i, j, k \\ \sum_{i,j,k} x_{ijk} \cdot t_{ijk} \leq TW_j, & \forall j \in D \\ x_{ijk} \in \{0, 1\}, y_j \in \{0, 1\}, & \forall i, j, k \end{cases} \end{aligned} \quad (16)$$

This multi-objective model represents a typical multi-objective mixed-integer nonlinear programming (MINLP) problem, characterized by strong coupling between objective functions, a non-convex variable space, and a large number of discrete variables. To address these complexities, a Pareto-optimal multi-objective solution strategy is required to obtain a set of balanced scheduling schemes that effectively trade off between cost efficiency and service quality.

3.1.2. NSGA-II Algorithm

The NSGA-II algorithm [31] is a classical multi-objective evolutionary optimization method. Its core lies in constructing a solution set that approximates the true Pareto front through fast non-dominated sorting and a crowding distance control mechanism, thereby achieving a well-balanced trade-off between solution diversity and convergence. The standard procedure of NSGA-II is as follows:

(1) Initialization

Randomly generate an initial population P_0 of size N and compute the multi-objective fitness value $F(x) = [f_1(x), f_2(x), \dots, f_m(x)]$ for each individual.

(2) Non-dominated sorting

Sort all individuals in the population based on dominance relations. For any two individuals x_i and x_j , if $\forall k \in \{1, \dots, m\}$ and $f_k(x_i) \leq f_k(x_j)$, then x_i is said to dominate x_j . The first layer of non-dominated individuals forms the approximate Pareto front F_1 .

(3) Crowding distance calculation

Within each non-dominated front, calculate the crowding distance S_i of each individual x_i based on the normalized distance between its neighboring solutions in each objective:

$$S_i = \sum_{k=1}^m \frac{f_k^{(i+1)} - f_k^{(i-1)}}{f_k^{\max} - f_k^{\min}} \quad (17)$$

where f_k^i denotes the position of individual x_i in the sorted list for the k -th objective. The crowding distance reflects the density of surrounding solutions in the objective space.

(4) Selection

Use a binary tournament selection strategy based on rank (non-dominated layer) and crowding distance to generate a parent population P_{t+1} .

(5) Crossover and mutation

Apply simulated binary crossover and polynomial mutation operators to the selected parents to generate the offspring population P_t .

(6) Population update

Combine the current generation's parents P_{t+1} and offspring P_t to form a temporary population $P_t = P_{t+1} \cup P_t$. Perform non-dominated sorting and select the top N individuals to form the next generation P_{t+1} .

(7) Iteration

Repeat the process until a predefined termination condition is satisfied.

While NSGA-II demonstrates advantages in non-dominated sorting and diversity preservation, it still faces limitations when handling high-dimensional objective spaces or mixed problems with strong nonlinear coupling and integer constraints. These challenges include weak local search capability, tendency to converge to suboptimal solutions, diminished resolution of the Pareto front in high-dimensional objectives leading to poor coverage, and lack of problem-specific operator design, reducing generalizability [32]. To address these issues, this study develops an enhanced NSGA-II algorithm to improve overall solving efficiency and solution quality, thereby enabling efficient resolution and decision support for complex multi-objective scheduling problems.

3.1.3. ENSGA-II Algorithm

The proposed ENSGA-II introduces a series of targeted improvements over the classical NSGA-II framework, including heuristic insertion-based initial population construction, goal-oriented local search operators, and stage-wise evolutionary control mechanisms.

(1) Improved strategy I: Heuristic insertion-based initial population construction

To enhance the initialization efficiency and solution feasibility of NSGA-II, this study proposes a heuristic insertion-based method for initial population construction. This approach addresses the limitations of traditional random initialization in terms of constraint satisfaction and distribution coverage within the solution space. The core idea is to incrementally construct high-quality initial solutions under the constraint of path feasibility. The improved initialization process is described as follows:

1) A task $i \in I$ is randomly selected to form the initial path $R_0 = \{i\}$. Each path R represents a scheduling sequence executed by a specific UAV departing from a designated distribution center. An appropriate combination of distribution center $j \in J$ and UAV model $k \in K$ is assigned to the path to ensure the following constraint is satisfied:

$$D(j, i) + D(i, j) \leq L_{jk} \quad (18)$$

where L_{jk} denotes the maximum flight range of UAV model k deployed at distribution center j , and $D(i, j)$ is the Euclidean distance between nodes i and j .

2) For the remaining tasks $i_r \in I_r$, each task is tentatively inserted into any legal position l in the existing paths $R_c \in R$. After insertion, the feasibility of the updated path must be verified, particularly with respect to the task deadline $T_{\max}$. The updated path must satisfy:

$$\max_{i_r \in R_c} \{T_{i_r}^{\text{end}}\} \leq T_{\max} \quad (19)$$

where $T_{i_r}^{\text{end}}$ is the estimated completion time of task i_r under the current path configuration, and $T_{\max}$ denotes its maximum allowable completion time. If the condition is met, the task i_r is inserted at position l , and the path is updated as:

$$R_c \cup \{l|i_r\}, \quad \forall R_c \in R \quad (20)$$

3) If task i_r cannot be inserted into any existing path without violating constraints, a new distribution center $j^* \in J$ and UAV model $k^* \in K$ are selected, under the condition that the distance between the center and the task does not exceed the maximum flight range:

$$D(j^*, i_r) \leq L_{jk} \quad (21)$$

A new path is then constructed as:

$$R_{\text{new}} = \{i_r\} \quad (22)$$

4) The above process is iteratively executed until all tasks are successfully assigned, resulting in a set of feasible solutions X_i . To ensure structural diversity within the initial population, a uniqueness

check is performed after each new solution is generated. A solution is added to the initial population P_0 only if it satisfies:

$$P_0 \cup \{X_i\}, \quad \forall P_0 \in P \quad (23)$$

The resulting initial population P_0 thus consists of p structurally diverse and constraint-compliant scheduling solutions. By incorporating constraint-aware evaluation and iterative path insertion strategies, this initialization method significantly reduces the proportion of infeasible solutions and enhances structural diversity within the feasible solution space, thereby providing a high-quality basis for the subsequent multi-objective evolutionary process.

(2) Improved strategy II: Goal-oriented top-down and bottom-up search operators

To enhance both the global exploration ability and local convergence performance in the multi-objective optimization process, a goal-oriented, staged top-down and bottom-up search mechanism is proposed to improve the performance of genetic operators. This mechanism comprises two customized operators: a goal-oriented crossover operator and a goal-oriented mutation operator, which collectively enhance solution quality and diversity through a two-phase construction strategy.

1) Goal-oriented crossover operator

This operator constructs high-quality offspring solutions through a two-stage path recombination mechanism. The first stage focuses on identifying and extracting promising path segments from the parent individuals to build a partial, high-quality offspring; the second stage is responsible for embedding remaining unassigned tasks into the existing path structure to ensure solution completeness and feasibility.

First stage: Construction of partial offspring (path selection and information coordination). In this stage, a partial solution is constructed based on information from two parent solutions, with priority given to extracting potentially valuable delivery paths while coordinating task information across paths to form an initial offspring solution X_h . Initially, the offspring solution is set to an empty set, i.e., $X_h = \emptyset$. Let the two parent solutions be X_1 and X_2 , with corresponding numbers of delivery paths $|X_1|$ and $|X_2|$. Define the minimum and maximum number of paths as:

$$MINR = \min(|X_1|, |X_2|) \quad (24)$$

$$MAXR = \max(|X_1|, |X_2|) \quad (25)$$

Then, set the number of path extraction iterations k based on the current optimization objective. If the objective is to minimize UAV usage (f_1), set $k = MINR - 1$ to encourage path merging and reduce total UAV count. If the objective is minimizing economic cost (f_2) or delay time (f_3), set $k = MINR$. In each iteration, a parent solution X_i is randomly selected. For the selected parent, identify the most promising path R_p^* according to the objective function f_p . The selection strategy is defined as:

$$R_p^* = \begin{cases} \arg \max_{R_p \in X_i} |R_p|, & f_p = f_1 \\ \arg \min_{R_p \in X_i} \frac{C_{eco}(R_p)}{|R_p|}, & f_p = f_2 \\ \arg \min_{R_p \in X_i} \frac{1}{|R_p|} \sum_{i \in R_p} \max(0, t_i - d_i), & f_p = f_3 \end{cases} \quad (26)$$

where X_i denotes the currently selected parent solution and $R_p \in X_i$ represents the set of delivery paths within that solution, $|R_p|$ indicates the number of task points contained in path R_p . A lower average delay per task point in a path implies a higher preference under objective function f_3 . Once the optimal path R_p^* is identified, it is copied and added to the offspring solution X_h , and simultaneously removed from the current parent solution.

In the other parent solution, all task points $i \in R_p^*$ included in this selected path R_p^* undergo a coordinated operation, as follows:

The task point i is removed from its original path, after which its predecessor and successor nodes are directly reconnected to form a continuous path structure. Relevant attributes, such as path length,

payload, and time windows, are then updated to ensure the feasibility and consistency of the modified path. This procedure is iteratively executed for k rounds. If at any point during the iterations either parent solution no longer contains feasible paths, the phase is terminated prematurely. Ultimately, the number of paths in the partial offspring solution X_h satisfies $|X_h| \leq k$. However, some task points may remain unassigned and will be handled in the second phase.

Second stage: Insertion of remaining customer tasks. In the partial offspring solution X_h constructed during the first stage, only a subset of delivery routes inherited from the parent solutions is retained, leaving some task points uncovered. Let the set of these unassigned task points be denoted as U . The aim of this phase is to insert all task points in U into the existing routes in a feasible and efficient manner, or to create new delivery routes when necessary, thereby forming a complete and feasible offspring solution.

Specifically, this stage operator processes the task points in U one by one. In each iteration, a randomly selected unassigned task point $i_r \in U$ is chosen, and its optimal insertion position within the current solution X_h is sought. The evaluation criterion for determining the best insertion position is based on the current optimization objective function f_p . If the current objective is to minimize the number of UAVs used (f_i), the primary principle is to avoid adding new routes. Among all existing routes, the one with the largest number of task points is given priority:

$$R_h^* = \arg \max_{R_h \in X_h} |R_h| \quad (27)$$

Then, within the selected route R_h^* , the operator sequentially searches for the first insertion position that satisfies all constraint conditions. Once a feasible position is found, the task point is inserted accordingly.

If the current objective is to minimize economic cost (f_2), the algorithm evaluates all routes and feasible insertion positions to calculate the incremental cost:

$$\Delta C_{\text{eco}}(R_h) = C(R_h, i_r) - C(R_h) \quad (28)$$

The task is inserted at the position with the smallest cost increase:

$$i^* = \arg \min \Delta C_{\text{eco}}(R_h) \quad (29)$$

For minimizing delay time (f_3), the delay increment is calculated as:

$$\Delta C_{\text{del}}(R_h) = C(R_h, i_r) - C(R_h) \quad (30)$$

Insertion occurs at the position minimizing delay increment:

$$i^* = \arg \min \Delta C_{\text{del}}(R_h) \quad (31)$$

If no feasible insertion exists, a new route R_{new} is created. First, assign the closest distribution center:

$$j_d = \arg \min_{j \in J} D(j, i_r) \quad (32)$$

Select a UAV type k_r satisfying range constraints:

$$\min_{j \in J} D(j_d, i_r) \leq \text{Range}(k_r) \quad (33)$$

Construct the new route:

$$R_{\text{new}} = \{j_d, i_r\} \quad (34)$$

This new route is then added to the current solution X_h . The above steps are repeated iteratively until all unassigned customer task points $i_r \in U$ have been successfully inserted, ultimately forming a complete and feasible offspring solution.

2) Goal-guided mutation operator

To enhance the population's adaptability and responsiveness during the multi-objective evolutionary process, this study designs two types of mutation operators with objective-awareness mechanisms. These operators apply perturbations and structured repairs tailored to different path characteristics and optimization objectives, such as the number of drones used, economic cost, and service delay time, thereby strengthening the algorithm's capability to balance multiple objectives.

Mutation operator I: Reconstruction and reinsertion of delayed paths (trigger probability $P=0.25$)

This mutation operator aims to identify and adjust paths containing delayed tasks in the current solution to reduce path-layer delay risks, thereby improving the timeliness and feasibility of the scheduling scheme. The specific procedure is as follows:

First, traverse all delivery paths R in the current solution and extract those containing at least one delayed task point to form a set of delayed paths:

$$R_{\text{delay}} = \{R_p \in R \mid \exists i \in R_p, t_i^c - d_i > 0\} \quad (35)$$

Secondly, for each path R_p in the set R_{delay} , remove all task points $i \in R_p$ where delays have occurred (i.e., tasks satisfying the corresponding delay condition), and denote these removed task points as the set U_{delay} . It is noteworthy that the original distribution center and drone model of the path remain unchanged. For each removed task point $i \in U_{\text{delay}}$, an attempt is made to reinsert it into the remaining routes of the current solution. The insertion position must satisfy the following conditions:

$$t_i^c \leq d_i, \forall i \in R_p, R_p \in R \quad (36)$$

Eq. (36) (now without numbering) ensures that, after insertion, the service completion time of all tasks on the path does not exceed the upper bound of their respective time windows. If no feasible insertion position can be found in the existing routes for the task point, a new path R_{new} is created for task i , starting from its original distribution center. The algorithm then searches for the optimal insertion position within the new path or its neighboring routes. The criterion for selecting the insertion position is to minimize the total service delay of the path, ideally achieving zero, while satisfying all constraints. If the insertion conditions remain unmet, additional backup paths are constructed, and insertion attempts continue iteratively until all delayed task points are successfully reassigned. This strategy effectively reduces the overall delay layer of the solution, providing a target-oriented proactive intervention mechanism that facilitates precise regulation of delay metrics during the multi-objective evolutionary process.

Mutation operator II: Destruction-reconstruction mechanism (trigger probability $P=0.75$).

This operator integrates dynamic preferences of multiple objective functions to achieve targeted perturbation and reconstruction of path structures, thereby enhancing the local feasibility repair capability and global exploration diversity within the solution space. The mechanism primarily consists of two stages:

In the destruction stage, first, the set U of unassigned task points is constructed. According to the dynamic preference of the current optimization objective function f_p , different destruction strategies are selected:

When the optimization objective is to minimize the number of drones used (f_1), priority is given to selecting the path with the fewest task points, and all tasks on this path are moved into the set U . If, after removing tasks from the path, its structure no longer satisfies feasibility constraints, it must be divided and reconstructed into several feasible sub-paths:

$$R_p^* = \arg \min_{R_p \in R} |R_p| \quad (37)$$

When the optimization objective is to minimize economic cost (f_2), two mutually exclusive strategies are employed and executed with probabilities of 70% and 30% respectively: The first is "random path partial removal," in which a path is randomly selected from the set of paths, and n task points are randomly removed from it ($n \in \mathbb{N}, |R_p|$). The second is "iterative random removal," where a maximum iteration number $T_{\max}$ is set, and in each iteration, a feasible path is randomly selected and one task point is removed.

When the optimization objective is to minimize delay time (f_3), two mutually exclusive strategies are also used: The first is "high-delay path partial removal," which selects the path with the largest total delay:

$$R_p^* = \arg \max_{R_p \in R} \sum_{i \in R_p} \max(0, t_i^c - d_i) \quad (38)$$

Then, randomly removes several task points from it. The second is "iterative random removal," where removal operations are randomly performed only on customer points in paths that contain service delays.

After completing the destruction phase, the reconstruction phase begins, where task points in the set U are inserted one by one. In each iteration, a task point i_r is randomly selected from U , and its optimal insertion position in the existing or new routes is determined based on the current optimization objective function:

If the objective function is to minimize the number of drones used (f_1), priority is given to the earliest feasible position where the insertion does not cause service delays for existing task points. If no such position exists, the last feasible position is chosen to maximize the insertion flexibility of the route. If the objective function is to minimize economic cost (f_2), the position causing the smallest increase in cost after insertion is selected. If the objective function is to minimize delay time (f_3), the position resulting in the minimum increase in total route delay is selected. If the current task point cannot be inserted into any existing route, a new route is created according to the following rule: the nearest distribution center in terms of Euclidean distance to task point i_r is selected as the starting point:

$$j_d = \arg \min_{j \in J} D(i_r, j) \quad (39)$$

Next, assign a UAV model $k_i \in K$ that meets the current task's service range requirements, which must satisfy the following range constraint:

$$\min_{j_d \in J} D(i_r, j_d) \leq \text{Range}(k_i) \quad (40)$$

Finally, during the population update process, this study adopts a "parent plus offspring" merging strategy to construct an intermediate population of size $2N$, and introduces an elitism-based nondominated sorting mechanism to ensure the continuous preservation of high-quality solutions. The proposed goal-oriented bi-directional search improvement operator strategy integrates structured crossover and mutation operations, enabling directional guidance and local fine-tuning reconstruction for different optimization objectives while maintaining solution feasibility. This strategy, combined with a staged evolutionary mechanism, achieves a balance between global exploration and local exploitation, significantly enhancing solution diversity and the capability to obtain high-quality solutions. It provides efficient and stable optimization support and an algorithmic foundation for multi-UAV task allocation problems. The pseudocode of the proposed ENSGA-II algorithm is presented as follows.

Algorithm 1 Enhanced NSGA-II Algorithm**Input:** Task set I , Depot set J , UAV model set K , Population size N , Maximum generations $G_{\max}$ **Output:** Pareto-optimal solution set X_i

```

1: Begin
2: initialize  $P_0 \leftarrow \emptyset$ 
3: while  $|P_0| < N$  do
4:   randomly select depot  $j$ , UAV type  $k$ 
5:   construct route set  $R$  by inserting tasks  $i \in I$  greedily, s.t.  $D(j, i) + D(i, j) \leq L_k$ 
6:   if insertion fails, reassign  $(j^*, k^*)$  and retry
7:   encode solution  $X$  from route set  $R$ 
8:   if  $X \notin P_0$  then
9:      $P_0 \leftarrow P_0 \cup \{X\}$ 
10: evaluate  $P_0$  using objective functions
11:  $t \leftarrow 0$ 
12: while  $t < G_{\max}$  do
13:    $Q_t \leftarrow \emptyset$ 
14:   for each parent pair  $(X_a, X_b) \in P_t$  do
15:     extract elite route  $R_a \in X_a$  and  $R_b \in X_b$ 
16:     initialize offspring  $X_c \leftarrow R_a \cup R_b$ 
17:     remove duplicate tasks in  $R_b$  already included from  $R_a$ 
18:      $I_r \leftarrow I \setminus \text{tasks}(X_c)$ 
19:     for each task  $i \in I_r$  do
20:       for each route  $R \in X_c$  do
21:         for each feasible position  $l$  in  $R$  do
22:           tentatively insert  $i$  at  $l$ , evaluate cost and constraint violations
23:           if time window, energy, payload constraints satisfied then
24:             if  $\text{cost}(R \text{ with } i) < \text{minCost}$  then
25:               update  $\text{minCost}$ ,  $\text{bestRoute}$ ,  $\text{bestPos}$ 
26:           if  $\text{bestRoute} \neq \emptyset$  then
27:             insert  $i$  into  $\text{bestRoute}$  at  $\text{bestPos}$ 
28:           else
29:             randomly select depot  $j^*$ , UAV  $k^*$  such that  $D(j^*, i) \leq L_{k^*}$ 
30:             create route  $R_{\text{new}} = \{i\}$  and add to  $X_c$ 
31:   add feasible offspring  $X_c$  to  $Q_t$ 
32:   for each solution  $X \in Q_t$  do
33:     if  $\phi(t) = \text{Exploration}$  then
34:       randomly select route  $R \in X$ 
35:       remove a subsegment  $R_s$ ,  $I_r \leftarrow R_s$ 
36:       shuffle  $I_r$ , reinsert into  $X$  using greedy insertion
37:       with probability  $p$ , change depot or UAV for a random route
38:     else
39:       identify critical route  $R^*$ 
40:       apply intra-route 2-opt or task swap to  $R^*$ 
41:       reassign 1–2 tasks from  $R^*$  to other routes to balance load
42:       if total cost reduced and constraints satisfied, accept modification
43:    $R_t \leftarrow P_t \cup Q_t$ 
44:   apply non-dominated sorting and crowding distance selection
45:   select top  $N$  individuals  $\Rightarrow P_{t+1}$ 
46:    $t \leftarrow t + 1$ 
47: return final Pareto front  $X_i \subseteq P_{G_{\max}}$ 

```

3.2. Lower-Layer Path Planning Model and Algorithm

Following the determination of the initial routes through the upper-layer task allocation, the lower-layer path planning model focuses on the detailed point-to-point trajectory design for unmanned aerial vehicles (UAVs). This process comprehensively accounts for path length, the cost of traversing

hazardous areas, and the cost associated with changes in flight altitude, while incorporating relevant flight constraints.

3.2.1. Mathematical Model

(1) Decision variables

In the lower-layer path planning model, to accurately characterize the trajectory layerization behavior of a multi-UAV system during task execution, the following sets and decision variables are defined. First, let the task node set $\mathcal{I}$ represent the departure and destination nodes for each delivery task assigned in the upper-layer scheduling model, satisfying the condition $|\mathcal{I}| \geq 2$. The flyable space grid set G is used to discretize the three-dimensional airspace, where each grid $g \in G$ contains geographical and flight-related attributes, such as risk coefficients and altitude information. The time step set T represents the sequence of discrete time intervals used to capture the dynamic evolution of the path generation process. Based on the above sets, main decision variables for describing path planning are defined as follows:

1) State variables:

$$x_{g,t} = \begin{cases} 1, & \text{if UAV is in grid } g \text{ at time step } t \\ 0, & \text{otherwise} \end{cases} \quad (41)$$

$$z_{g,g',t} = \begin{cases} 1, & \text{if UAV moves from grid } g \text{ to grid } g' \text{ at time step } t \\ 0, & \text{otherwise} \end{cases} \quad (42)$$

2) Continuous variables:

Let $h_{g,t} \in \mathbb{R}^+$ denote the flight altitude (in meters) of UAV when located in grid g at time step t . Let $v_{g,g',t} \in \mathbb{R}^+$ represent the climb rate (in meters per second) of UAV when moving from grid g to grid g' at time step t . The above sets and decision variables together form the fundamental description of the lower-layer path planning problem, providing a modeling foundation for subsequent trajectory generation and layerization.

(2) Objective function

In the lower-layer multi-UAV path planning model, three sub-objectives are considered: path length, flight risk, and altitude variation cost.

1) Flight path length:

Path length serves as a critical indicator of mission execution efficiency. Minimizing the total flight path length can effectively improve delivery timeliness and reduce energy consumption:

$$\min L = \sum_{t \in T} \sum_{g \in G} \sum_{g' \in G} D_{g,g'} \cdot z_{g,g',t} \quad (43)$$

where $D_{g,g'}$ represents the Euclidean distance between grid cell g and grid cell g' . By summing the distances corresponding to actual grid transitions across all time steps, the total path length can be quantified.

2) Grid risk exposure:

To enhance mission safety, the planned path should avoid areas with high hazard coefficients, thereby reducing the risk from environmental factors. Thus, minimizing the total risk exposure becomes a secondary objective:

$$\min B = \sum_{t \in T} \sum_{g \in G} r_g \cdot x_{g,t} \quad (44)$$

where r_g represents the hazard coefficient of grid g , a predefined parameter.

3) Altitude variation cost:

Frequent altitude adjustments during UAV flight increase energy consumption and flight control complexity. Therefore, minimizing altitude variation cost is necessary to maintain energy efficiency and stability. The altitude variation cost is computed as:

$$\min H = \sum_{t \in T} \sum_{g \in G} \sum_{g' \in G} |\Delta h_{g,g'}| \cdot z_{g,g',t} \quad (45)$$

where $|\Delta h_{g,g'}|$ denotes the change in altitude when flying from grid g to grid g' . The weighted sum of absolute altitude differences quantifies the cost of altitude adjustments, thereby promoting smoother and more energy-efficient flight paths.

Additionally, this study adopts a hierarchical optimization strategy tailored to the distinct characteristics of decision-making at different layers. The upper-layer model focuses on global task allocation among UAVs, task locations, and distribution centers, involving multiple objectives with inherent conflicts, such as economic cost and task delay. Therefore, particular attention is given to the trade-offs among objectives. To this end, the upper-layer problem is addressed using a non-dominated sorting-based multi-objective optimization approach to construct a Pareto front solution set, accommodating diverse decision-making preferences and enhancing the adaptability and flexibility of scheduling strategies. In contrast, the lower-layer path planning problem features high dimensionality, stringent constraints, and complex combinatorial solution space. Directly applying multi-objective Pareto optimization in this context results in significant computational overhead and may hinder stable convergence. Therefore, after formulating the sub-objective functions at the lower layer, this study employs a weighted-sum method to transform the multi-objective problem into a single-objective one. This method not only enables the incorporation of decision preferences through weight coefficients but also facilitates integration with heuristic algorithms, thereby improving convergence performance and local search capability to efficiently obtain feasible solutions.

Let a_1, a_2, a_3 denote the weighting coefficients corresponding to path length, hazard exposure, and altitude variation cost, respectively, and satisfy the normalization constraint:

$$a_1 + a_2 + a_3 = 1 \quad (46)$$

The final integrated optimization objective function is:

$$\min f = a_1 \cdot L + a_2 \cdot B + a_3 \cdot H \quad (47)$$

By adjusting the values of the weight coefficients, the model can flexibly balance flight efficiency, safety, and energy stability according to specific application requirements, thereby ensuring optimal path planning performance for the multi-UAV system.

(3) Constraints

To ensure that the generated path planning solutions are compatible with real-world flight operations and task execution requirements, the following constraints are introduced:

1) Task start and end constraints:

Each delivery task must begin at a designated start grid and ultimately reach a specified end grid. This is formulated as:

$$\sum_{g \in G_{\text{start}}} x_{g,0} = 1 \quad (48)$$

$$\sum_{g \in G_{\text{end}}} x_{g,T} = 1 \quad (49)$$

where G_{start} and G_{end} represent the sets of start and end grids for task i , respectively. These constraints ensure that the UAV trajectory possesses well-defined departure and arrival nodes.

2) Continuous movement constraint:

A UAV must move between adjacent grids in consecutive time steps. That is, if it is located in grid g at time step t , it must move to a neighboring grid g' at time step $t + 1$. The constraint is expressed as:

$$x_{g,t} - \sum_{g' \in G(g)} z_{g,g',t} = 0, \quad \forall g \in G, \forall t \in T \quad (50)$$

where $G(g)$ denotes the set of grids adjacent to grid g . This ensures trajectory continuity and physical feasibility of flight paths.

3) Flight altitude constraints:

At any time step, the UAV must operate within a predefined altitude range to comply with airspace regulations and ensure flight safety:

$$h_{\min} \leq h_{g,t} \leq h_{\max}, \quad \forall g \in G, \forall t \in T \quad (51)$$

where $h_{\min}$ and $h_{\max}$ are the minimum and maximum allowable flight altitudes, respectively.

4) Maximum climb angle constraint:

During continuous operations, the UAV's climbing or descending angle must remain within the allowable maximum to ensure flight stability and safety. This constraint is expressed as:

$$\theta_t = \arctan\left(\frac{z_{t+1} - z_t}{\sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2}}\right) \leq \theta_{\max}, \quad \forall t \in T \quad (52)$$

where (x_t, y_t, z_t) and $(x_{t+1}, y_{t+1}, z_{t+1})$ denote the UAV's spatial coordinates at consecutive time steps t and $t + 1$, and $\theta_{\max}$ is the maximum allowable climb angle.

5) Obstacle and no-fly zone avoidance constraint:

The UAV's path must not intersect any predefined obstacles or no-fly zones. This constraint is given by:

$$x_{g,t} = 0, \quad \forall g \in G_{f0}, \forall t \in T \quad (53)$$

where G_{f0} represents the set of impassable or restricted grid cells.

In summary, at the lower-layer path planning stage, a comprehensive optimization model is established that simultaneously considers minimizing path length, reducing flight risk, and minimizing altitude variation costs. This model accounts for the spatial-temporal dynamics of UAV flight while ensuring physical feasibility and task completion. Due to the use of grid-based spatial modeling and discrete-time step planning, the problem scale increases significantly. As a result, the overall model exhibits characteristics of a strongly coupled, non-convex, and mixed-integer optimization problem, which increases computational complexity. Therefore, this study introduces an improved version of the traditional Particle Swarm Optimization (PSO) algorithm to efficiently generate feasible path planning decisions.

3.2.2. Standard PSO Algorithm

The Particle Swarm Optimization (PSO) algorithm [35] is inspired by the social foraging behavior of bird flocks. It utilizes information sharing and collaboration among individuals within the swarm to search for optimal solutions in a given space. Due to its simple structure, minimal parameter requirements, and ease of implementation, PSO has been widely applied in continuous optimization, combinatorial optimization, and multi-objective optimization problems.

In PSO, each potential solution is abstracted as a particle characterized by two key attributes: position and velocity, which correspond to the decision variable values and the direction of search, respectively. During each iteration, the velocity and position of each particle are updated based on its own historical best position (p_{best}) and the global best position (g_{best}) identified by the swarm. The update equations are defined as:

$$v_i^{t+1} = \omega \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i^{\text{best}} - x_i^t) + c_2 \cdot r_2 \cdot (g_i^{\text{best}} - x_i^t) \quad (54)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (55)$$

where v_i^t and x_i^t represent the velocity and position of particle i at iteration t . ω is the inertia weight, balancing the global and local search abilities of particles. c_1 and c_2 are the cognitive and social learning factors, respectively, which determine the attraction to p_{best} and g_{best} . r_1 and r_2 are random numbers within $[0, 1]$ that introduce stochasticity to enhance diversity.

The standard PSO algorithm leverages a simple and efficient mechanism that integrates both local and global information, enabling rapid convergence toward optimal solutions. However, when applied to high-dimensional, non-convex, strongly coupled, and constraint-intensive problems, it tends to suffer from reduced convergence speed, premature convergence to local optima, and insufficient search accuracy. Therefore, to address the specific characteristics of the proposed multi-UAV path planning model, it is necessary to introduce adaptive enhancements to the standard PSO framework in order to improve its solution capability and the quality of path generation.

3.2.3. Improved PSO Algorithm

(1) Improvement strategy 1: Chaotic initialization

To overcome the limited diversity and weak global search ability at the initial stage of traditional PSO, a chaotic initialization strategy [36] is introduced to enhance population diversity and global exploration. Chaotic sequences possess ergodicity, randomness, and determinism, effectively preventing premature convergence to local optima. This study adopts the Singer map for chaotic sequence generation. Compared to Logistic and Tent maps, the Singer map maintains chaotic behavior while offering a more uniform distribution, making it suitable for initialization in high-dimensional search spaces:

$$x_{n+1} = \mu \cdot (7.86x_n - 23.31x_n^2 + 28.75x_n^3 - 13.302875x_n^4), \quad x_n \in (0, 1) \quad (56)$$

where μ is the control parameter, typically set to 1 to ensure chaotic behavior. x_n is the initial value within the interval $(0, 1)$, used to generate a chaotic sequence $\{x_n\}_{n=1}^N$. This strategy ensures a more even distribution of the initial population in the solution space, improving solution diversity and global exploration capacity, and reducing the risk of premature convergence.

(2) Improvement strategy 2: Linearly adjusted acceleration coefficients and maximum velocity

In PSO algorithm, the acceleration coefficients c_1 and c_2 control the particles' tendencies to learn from personal and global best positions, respectively. Their values significantly influence algorithm performance. Traditional PSO often fixes them as constants, which cannot effectively balance global and local search in dynamic environments. Thus, a linear adjustment mechanism [37] is introduced to dynamically tune and so that particles transition from "exploration-oriented" to "exploitation-oriented" behaviors:

$$c_1(t) = c_{1,\text{max}} - (c_{1,\text{max}} - c_{1,\text{min}}) \cdot \frac{t}{T_{\text{max}}} \quad (57)$$

$$c_2(t) = c_{2,\text{min}} + (c_{2,\text{max}} - c_{2,\text{min}}) \cdot \frac{t}{T_{\text{max}}} \quad (58)$$

where t is the current iteration and T_{max} is the maximum number of iterations. $c_{1,\text{max}}$ and $c_{1,\text{min}}$ denote the upper and lower bounds for the individual learning coefficient, while $c_{2,\text{max}}$ and $c_{2,\text{min}}$ represent those for the social learning coefficient. In early iterations, a larger c_1 and smaller c_2 encourage exploration based on individual experience. In later iterations, c_1 decreases while c_2 increases, enhancing cooperative search and improving convergence speed and solution quality.

In addition, the maximum particle velocity v_{max} plays a critical role in constraining the step size of particles, directly affecting the search range and convergence speed. If v_{max} is too large, particles may overshoot promising regions, resulting in unstable convergence; if too small, particle movement becomes overly restricted, increasing the risk of local stagnation. To ensure a proper balance between

global exploration and local refinement, a linearly decreasing velocity strategy [38] is employed to reduce $v_{\max}$ as iterations progress. The adjustment rule is given by:

$$v_{\max}(t) = v_{\max}^p - (v_{\max}^p - v_{\min}^p) \cdot \frac{t}{T} \quad (59)$$

where $v_{\max}(t)$ is the maximum velocity at iteration t , $v_{\max}^p$ is the initial maximum velocity, $v_{\min}^p$ is the final minimum velocity.

(3) Improvement strategy 3: Particle position update based on random perturbation

In standard PSO, particle positions are typically updated based solely on velocity. However, for path planning problems characterized by discrete space modeling and strong non-convexity, such linear updates often lead to a homogeneous particle distribution, reducing diversity and search quality. To improve exploratory capability and help escape local optima, a random perturbation-based position update mechanism [39] is introduced on top of the standard update. Specifically, a controlled stochastic disturbance is added to the position update to enable fine-scale jumping behavior. The new position of a particle is calculated as:

$$p = p_{\text{large}} + \alpha \cdot r_3 \cdot (v_{\text{large}} - v_{\text{small}}) \quad (60)$$

where p_{large} is the position of a better-performing particle in the current iteration, v_{large} and v_{small} are large and small velocity values, α is a perturbation control factor (typically a small positive value < 1), and r_3 is a random number in $[0, 1]$.

(4) Improvement strategy 4: Adaptive adjustment of inertia weight

The inertia weight is a critical parameter in the particle swarm optimization (PSO) algorithm that regulates a particle's "memory" and guides its search behavior. The magnitude of the inertia weight directly influences the algorithm's balance between global exploration and local exploitation. A larger inertia weight facilitates broader exploration of the search space and enhances the algorithm's ability to escape local optima, whereas a smaller inertia weight favors finer local search, thus promoting convergence to optimal solutions in promising regions. To better accommodate the varying demands of different search phases, an adaptive inertia weight adjustment strategy [40] is introduced. This strategy employs a linearly decreasing inertia coefficient based on the iteration progress, enabling a dynamic transition in the search focus from global to local as the algorithm proceeds. The adjustment formula is expressed as follows:

$$\omega_{t+1} = \omega_t - \frac{0.75t^3}{T^3} (\omega_{\max} - \omega_{\min})^3 \quad (61)$$

where ω_t is the inertia weight at generation t , $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum weights, respectively. The pseudocode of the improved PSO algorithm is presented as follows.

Algorithm 2 Improved PSO

Input: Number of particles N , Maximum number of iterations T , Maximum and minimum acceleration coefficients c_1 and c_2 , Maximum and minimum velocities $v_{\max}$ and $v_{\min}$, Position update constant α , control parameter μ

Output: Optimal velocity v_{large} , Optimal position P_{large}

- 1: **Begin**
- 2: initialize the particle population using chaotic Singer mapping
- 3: **for** each $t = 1$ to T **do**
- 4: evaluate the fitness of each particle:

$$x_{n+1} = \mu \cdot (7.86x_n - 23.31x_n^2 + 28.75x_n^3 - 13.302875x_n^4)$$

- 5: update acceleration coefficients using linear adjustment:

$$c_1(t) = c_{1,\max} - (c_{1,\max} - c_{1,\min}) \cdot \frac{t}{T}, \quad c_2(t) = c_{2,\min} + (c_{2,\max} - c_{2,\min}) \cdot \frac{t}{T}$$

- 6: adjust maximum velocity linearly:

$$v_{\max}(t) = v_{\max}^p - (v_{\max}^p - v_{\min}^p) \cdot \frac{t}{T}$$

- 7: execute standard PSO operations and sort particles by fitness
- 8: **for** each particle $i = 1$ to N **do**
- 9: update position using:

$$p = P_{\text{large}} + \alpha \cdot r \cdot (v_{\text{large}} - v_{\text{small}})$$

- 10: update velocity using:

$$v_i^{t+1} = \omega \cdot v_i^t + c_1 \cdot r_1 \cdot (P_i^{\text{best}} - x_i^t) + c_2 \cdot r_2 \cdot (g^{\text{best}} - x_i^t)$$

- 11: adjust inertia weight adaptively:

$$\omega_t = \omega_{\max} - \frac{0.75t^3}{T^3}(\omega_{\max} - \omega_{\min})$$

- 12: **return** v_{large} and P_{large}
- 13: **End**

4. Simulation and Analysis

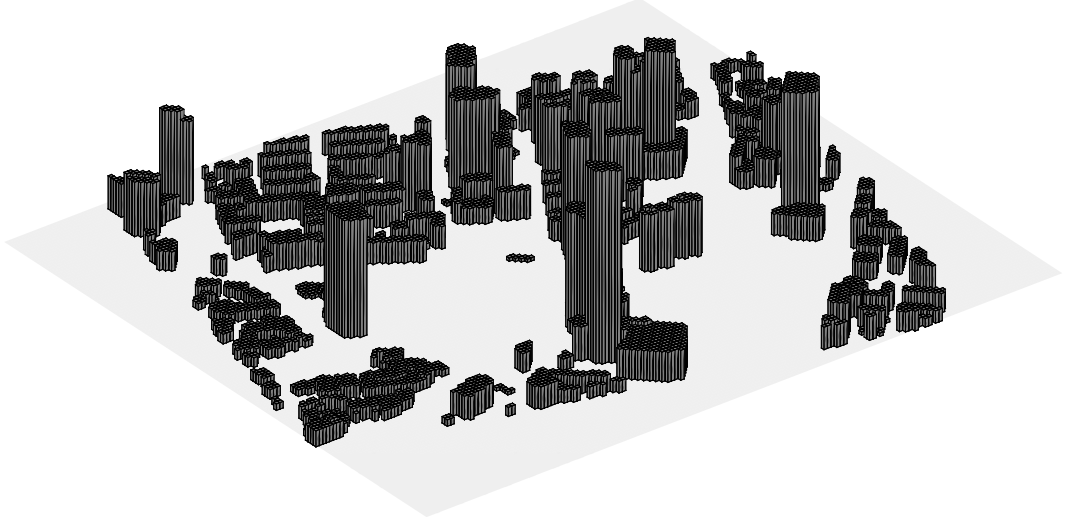
4.1. Simulation Environment

4.1.1. Environment Modeling

This study selects the Wudadao (Five Avenues) architectural district in Heping District, Tianjin, China, as the background for environmental modeling. As illustrated in Fig. 5, Figure 5(a) presents the satellite imagery of the street network, while Figure 5(b) shows the simulated environment constructed using a grid-based method, in which each grid cell corresponds to 0.1 kilometers in the actual geographic space.



(a) Street Satellite Image



(b) Environmental Simulation Map

Figure 5. Schematic diagram of UAV environmental modeling.

4.1.2. Experimental Setup

All simulations are conducted under the Windows 10 operating system, utilizing an Intel Core i7-13600K processor, an NVIDIA GeForce RTX 3060 graphics card with 12 GB VRAM, and 16 GB DDR4 memory. Model development and simulation implementation are performed in the PyCharm integrated development environment, using Python 3.9. This experimental configuration provides sufficient computational resources to ensure algorithmic efficiency and high-quality solution sets for high-dimensional optimization problems.

4.1.3. Parameter Configuration

In the simulation analysis, the number of logistics depots is set to 3. The detailed configuration of depot-related parameters is provided in Table 1. Two DJI transport UAV models with distinct performance characteristics are employed, and their specifications are listed in Table 7. The number of task nodes is set to 50, with time window constraints defined as [0,90]. Furthermore, in algorithm validation experiments, the number of task nodes is extended to 100 to fully evaluate the applicability and scalability of different algorithms under high-dimensional task conditions. The configuration of parameters related to the bi-layer optimization model is summarized in Table 3.

Table 1. Distribution center parameter settings.

Distribution center No.	Task coordinates	Time window constraints/h
J1	(40,125,120)	8
J2	(50,40,120)	8
J3	(125,40,120)	8

Table 2. UAV parameter settings.

Parameter name	Parameter setting value	
UAV type	K1	K2
Flight speed/km·h ⁻¹	54	30
Max flight distance/km	40	30
Max payload/kg	50	150
Max flight time/min	44	60

Table 3. Parameter settings related to models and algorithms.

Parameter name	Parameter value	Parameter name	Parameter value
a_1	0.4	a_2	0.5
a_3	0.1	l_{grid}	5
h_{min}	120	h_{max}	200
θ_{max}	90	r_1	rand[0,1]
r_2	rand[0,1]	r_3	rand[0,1]
Maximum number of iterations G_{max} (ENSGA-II)	100	Maximum number of iterations G_{max} (IPSO)	50
Population N (ENSGA-II)	250	Population N (IPSO)	100

4.2. Experimental Results Analysis

4.2.1. Task Allocation Results Analysis

Figure 6 presents the final non-dominated solution set obtained based on the multi-objective optimization model, providing a comprehensive evaluation of task allocation schemes across three dimensions: economic cost, the number of UAVs scheduled, and time window violation duration. In the figure, green points represent the feasible solution set satisfying all constraints.

As illustrated in Figure 6, the constructed upper-layer multi-objective scheduling model and algorithm effectively reflect the trade-offs among the number of scheduled UAVs, service timeliness, and economic efficiency. The presented non-dominated solution set exhibits a typical concave distribution pattern, indicating that improvement in one objective often requires sacrificing performance in others. This aligns with the systemic resource constraints and dynamic priority adjustment mechanisms inherent in multi-UAV task allocation scenarios. The solution set shows good distribution density and boundary coverage within the three-dimensional objective space, demonstrating stable convergence and extensive search capability of the model without clustering or premature convergence. These results directly validate the proposed model and algorithm’s global adaptability and structural stability in multi-objective scheduling contexts, while providing diversified alternative solutions for practical applications with varying scheduling preferences.

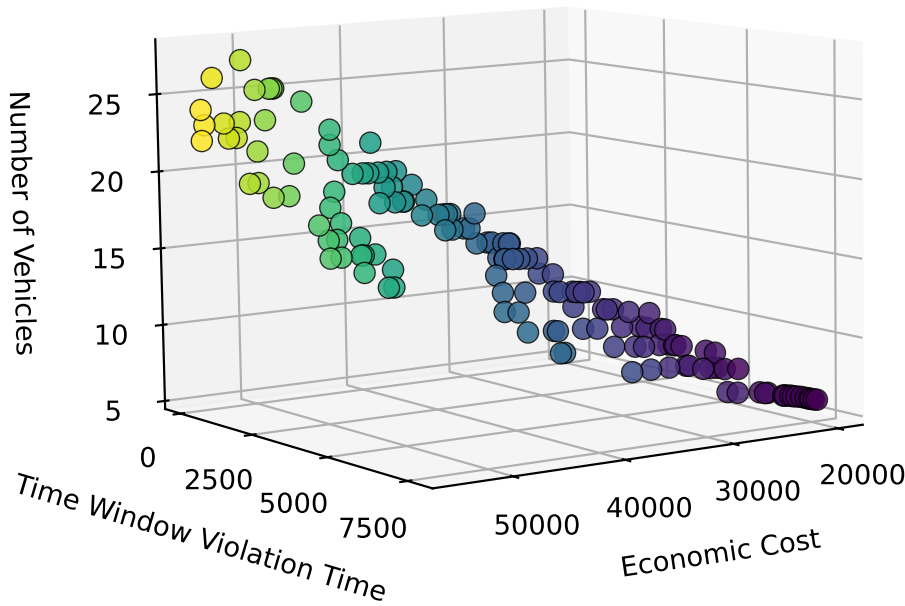


Figure 6. Pareto front of the upper-layer model.

To further elucidate the internal trade-off relationships of the proposed multi-objective optimization model and algorithm, the three-objective Pareto front is projected onto three representative two-dimensional objective spaces for visualization, as shown in Figure 7.

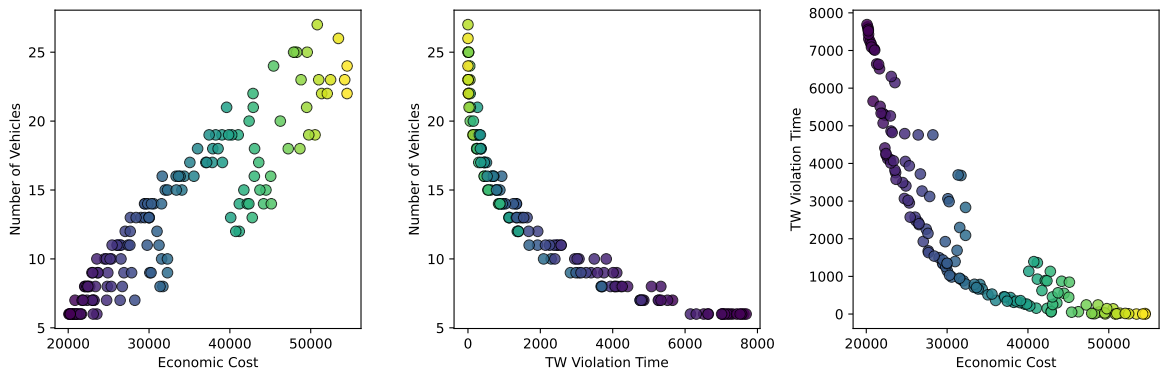


Figure 7. Decomposed two-objective Pareto front:(a) UAVs vs Costs;(b) UAVs vs Violation time;(c) Violation time vs Costs.

Figure 7(a) depicts the relationship between economic cost and the number of UAVs utilized, showing a pronounced positive correlation trend. This indicates that, in most cases, reducing economic expenditure requires scheduling fewer UAVs, a resource constraint particularly evident in multi-depot environments. The result highlights the coordination capacity of the task allocation scheme between resource configuration and cost control. Figure 7(b) illustrates the Pareto distribution between task delay time and UAV quantity, displaying a characteristic power-law decay. It is observable that when UAV quantity is limited, overall system delay significantly increases, whereas appropriate augmentation of UAVs effectively mitigates task time conflicts, enhancing scheduling timeliness. However, beyond a certain UAV threshold, marginal benefits diminish, reflecting saturation in resource allocation for timeliness improvement. Figure 7(c) presents the relationship between economic cost

and task delay time, exhibiting a typical nonlinear negative correlation. Low-cost solutions are often accompanied by higher delay durations, while achieving minimal delay incurs substantial cost penalties. This vividly demonstrates the inherent conflict between economy and timeliness, and confirms that the proposed algorithm effectively approximates the Pareto front among competing objectives while maintaining solution diversity. The distribution characteristics of the three bi-objective Pareto fronts validate the adaptability and flexibility of the proposed model and algorithm in handling high-dimensional multi-objective conflicts.

To systematically assess the overall performance of the constructed multi-objective bi-layer collaborative task planning model and the improved optimization algorithm in practical scenarios, three representative planning schemes are selected from the obtained Pareto front for quantitative comparative analysis. The specific optimization results are summarized in Table 4. Scheme 1 corresponds to the initial non-dominated solution on the Pareto front and serves as a baseline for performance evaluation. Schemes 2 and 3 represent feasible plans achieving balanced overall improvements by comprehensively weighing UAV scheduling quantity, economic cost, and violation time under conflicting objectives.

Table 4. Comparative optimization results of different planning schemes.

Schemes	Number of UAVs	Violation time/s	Economic cost /CNY
1	23	0.00	53798.07
2	19	331.79	38660.01
3	16	679.47	33267.61

As shown in Table 4, Scheme 1 aims to completely avoid time window violations. Its planning structure embodies a resource-redundant configuration centered on optimal task timeliness, scheduling 23 UAVs to ensure zero violations throughout the entire task duration. The economic cost reaches 53798.07 CNY, exhibiting a typical "high resource-high cost-zero delay" characteristic. Although this scheme achieves the optimal boundary in service quality, it suffers from high redundancy in resource input and relatively weak economic sustainability. Building upon this, Scheme 2 moderately relaxes the time window constraints under the premise that delivery efficiency remains acceptable. The number of scheduled UAVs is reduced from 23 in Scheme 1 to 19, a 17.4% reduction, significantly enhancing resource utilization efficiency. Correspondingly, the time window violation increases to 331.79 seconds, slightly higher than the zero seconds in Scheme 1 but still well below most planning tolerance thresholds, indicating controllable delay layers. Meanwhile, the economic cost decreases from 53798.07 CNY to 38660.01 CNY, achieving a 28.1% cost reduction. This dual optimization of resource input and cost is realized while maintaining basic timeliness. These results demonstrate that the model effectively balances delivery efficiency and economic performance, suitable for operational scenarios that are cost-sensitive yet have relatively flexible timeliness requirements. Furthermore, Scheme 3 emphasizes pushing the system’s operational limits under tightened resource constraints and cost reduction conditions. This scheme schedules only 16 UAVs, a 30.4% reduction compared to Scheme 1, demonstrating strong potential for resource compression. The economic cost further decreases to 33267.61 CNY, a 38.2% reduction relative to Scheme 1, representing the lowest value among the three schemes. Although the violation time increases to 679.47 seconds, compared to zero delay in Scheme 1, the increase remains within acceptable delivery service limits. The delay accounts for only a small portion of the total task cycle time, indicating that the system maintains stable task completion capability under high-intensity resource constraints. This scheme verifies the model and algorithm’s high robustness, cost adaptability, and structural control capability under extreme scheduling conditions.

Overall, the results indicate that the proposed model and algorithm possess strong capabilities in balancing objectives and resource allocation. They can provide high-quality feasible solutions adapted to different scheduling preferences under multiple constraints. The computed non-dominated

solutions form a clear multi-objective cooperative frontier, validating the stability and robustness of the algorithm.

As shown in Table 4, Scheme 1 aims to completely avoid time window violations. Its planning structure embodies a resource-redundant configuration centered on optimal task timeliness, scheduling 23 UAVs to ensure zero violations throughout the entire task duration. The economic cost reaches 53798.07 CNY, exhibiting a typical "high resource-high cost-zero delay" characteristic. Although this scheme achieves the optimal boundary in service quality, it suffers from high redundancy in resource input and relatively weak economic sustainability. Building upon this, Scheme 2 moderately relaxes the time window constraints under the premise that delivery efficiency remains acceptable. The number of scheduled UAVs is reduced from 23 in Scheme 1 to 19, a 17.4% reduction, significantly enhancing resource utilization efficiency. Correspondingly, the time window violation increases to 331.79 seconds, slightly higher than the zero seconds in Scheme 1 but still well below most planning tolerance thresholds, indicating controllable delay layers. Meanwhile, the economic cost decreases from 53798.07 CNY to 38660.01 CNY, achieving a 28.1% cost reduction. This dual optimization of resource input and cost is realized while maintaining basic timeliness. These results demonstrate that the model effectively balances delivery efficiency and economic performance, suitable for operational scenarios that are cost-sensitive yet have relatively flexible timeliness requirements. Furthermore, Scheme 3 emphasizes pushing the system's operational limits under tightened resource constraints and cost reduction conditions. This scheme schedules only 16 UAVs, a 30.4% reduction compared to Scheme 1, demonstrating strong potential for resource compression. The economic cost further decreases to 33267.61 CNY, a 38.2% reduction relative to Scheme 1, representing the lowest value among the three schemes. Although the violation time increases to 679.47 seconds, compared to zero delay in Scheme 1, the increase remains within acceptable delivery service limits. The delay accounts for only a small portion of the total task cycle time, indicating that the system maintains stable task completion capability under high-intensity resource constraints. This scheme verifies the model and algorithm's high robustness, cost adaptability, and structural control capability under extreme scheduling conditions.

Overall, the results indicate that the proposed model and algorithm possess strong capabilities in balancing objectives and resource allocation. They can provide high-quality feasible solutions adapted to different scheduling preferences under multiple constraints. The computed non-dominated solutions form a clear multi-objective cooperative frontier, validating the stability and robustness of the algorithm.

4.2.2. Path Planning Results Analysis

To validate the adaptability and effectiveness of the constructed lower-layer UAV path planning model within the multi-distribution-center collaborative scheduling environment, this study systematically organizes the results of the lower-layer path planning tasks and compiles a UAV delivery task table. Table 5 presents the number of UAVs scheduled by each of the three distribution centers along with their corresponding delivery point assignments, intuitively reflecting the model's capability in task allocation and path planning under multi-center coordination.

As shown in Table 5, this study adopts a differentiated UAV task allocation strategy tailored to the spatial heterogeneity of each distribution center's coverage area. Specifically, Distribution center 3 is responsible for the largest number of delivery points (a total of 23), with a clearly dispersed and wide-area spatial distribution. To effectively address the path complexity caused by this high degree of dispersion and to achieve a balanced task load, the system assigns the highest number of UAVs (six in total) to this center. Based on spatial partitioning principles, this dense scheduling strategy divides the widely distributed delivery points into multiple operational units, thereby significantly reducing the average number of service points and flight path length per UAV. This not only ensures complete task coverage but also optimizes the overall operating cost and service timeliness. In contrast, Distribution center 2 is responsible for fewer delivery points (a total of 10), with a highly clustered spatial distribution. For instance, nearby points such as 22, 42, 43, 52 and 17, 18, 26 are assigned to

the same UAV. This natural spatial proximity allows the system to achieve efficient coverage with the smallest number of UAVs (only three). The centralized scheduling strategy leverages geographic aggregation effects, enabling a single flight mission to cover multiple nearby targets. This approach significantly reduces system dispatching costs while improving resource utilization. Distribution center 1 falls between these two cases. It manages 18 delivery points, and its spatial configuration features a combination of local clustering and moderate dispersion. For example, points 36, 37, and 38 form a significant cluster, while points 11, 13, 15, 35, and 45 are more dispersed. The system assigns six UAVs to this center, with the number of service points per UAV varying from one to five. This allocation pattern demonstrates the model’s flexibility in task packaging and spatial resource coordination. For delivery points in close proximity, the system tends to group them into a single dispatch to improve path efficiency; for isolated or off-route points (e.g., point 20), the system adopts a low-density or single-point delivery strategy. This ensures full coverage while achieving dynamic matching between task load and resource allocation, further validating the adaptability and optimization performance of the proposed lower-layer path planning model under scenarios involving regional spatial heterogeneity.

Table 5. UAV delivery task table in lower - layer path planning.

Distribution center No.	UAV No.	Assigned delivery point No.
1	1	20
	2	37,38,36
	3	11,13,15,45,35
	4	48,40,39
	5	34,33,41
	6	19,49
2	1	42,43,52,22
	2	17,18,26
	3	31,50,51
3	1	6,4,12,30
	2	16,28,21,5,9
	3	27,8,24
	4	14,32
	5	47,44,46
	6	10,7,23,29,3,25

Figure 8 illustrates the UAV path planning results corresponding to the delivery task table in Table 5. These results are generated by the lower-layer path planning model and its associated algorithm. In the figure, blue dots represent delivery point locations, square nodes indicate dispatch centers or UAV takeoff/landing points, avd_{ob} represents obstacle avoidance waypoints along UAV paths, and the background heatmap shows obstacle intensity, where brighter areas denote higher terrain complexity or flight cost. The lines connecting the nodes represent UAV flight paths.

As shown in Figure 8, the overall distribution of flight paths reveals a clear spatial partitioning pattern across UAV task areas, with no significant overlap or intersection among routes. This confirms the effectiveness of the proposed lower-layer path planning algorithm in achieving rational task load allocation. Tasks are efficiently distributed within the sub-regions managed by each distribution center, demonstrating a coordinated coupling between task area partitioning and route planning. Moreover, the majority of UAV flight paths proactively avoid high-risk zones (e.g., red triangles indicating rerouted waypoints for obstacle avoidance), indicating that the proposed model and algorithm are well adapted to flight safety constraints. The paths remain overall compact, with no redundant routes or loops, verifying the model’s capability to generate efficient route connections under multiple constraints such as time windows, flight distance, and UAV endurance. This ensures a coordinated integration of global scheduling objectives and local route optimization.

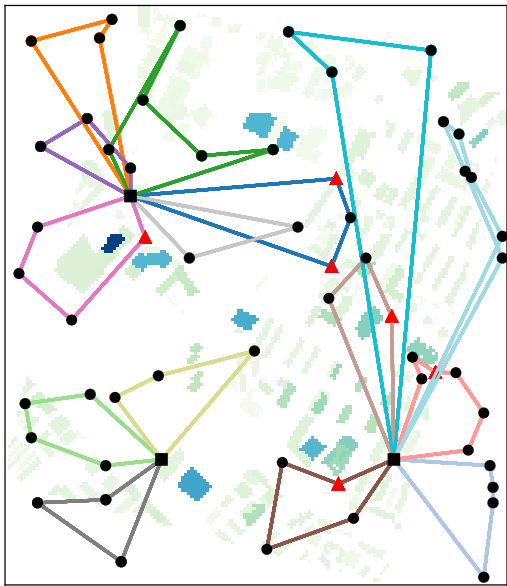


Figure 8. UAV path planning results of the lower-layer model.

In summary, the proposed two-layer collaborative planning model establishes a robust linkage between task allocation and path planning. The upper-layer strategy effectively guides the direction of lower-layer path evolution, while the lower-layer routes are highly responsive to upper-layer requirements and constraints. This mutual coordination achieves a global balance between task completeness and cost control.

4.3. Algorithm Performance Analysis

As the performance of the path planning algorithm has already been analyzed in detail in reference [38], this section focuses solely on the performance evaluation of the task allocation algorithm.

4.3.1. Analysis of the ENSGA-II Algorithm

To systematically evaluate the search performance and convergence behavior of the proposed ENSGA-II algorithm in solving multi-objective optimization problems, this section considers three objectives: economic cost, time window violation time, and the number of UAVs. The convergence curves over the course of iterations are illustrated in Figure 9. The analysis is conducted from three perspectives, convergence speed, fluctuation behavior, and convergence accuracy, to validate the algorithm’s adaptability and optimization potential.

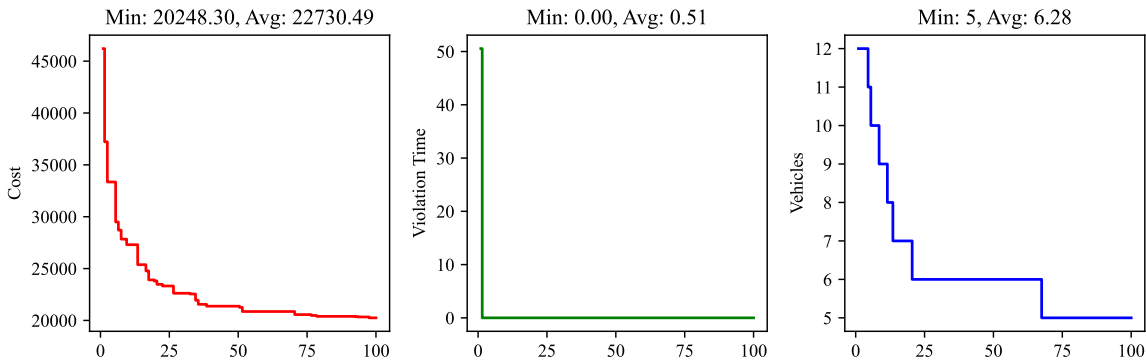


Figure 9. Convergence process of the algorithm on different optimization objectives:(a) Convergence curve of economic cost;(b) Convergence curve of time window violation;(c) Convergence curve of number of UAVs.

As shown in Figure 9(a), regarding the economic cost, the algorithm demonstrates strong search capability from the initial stages, with the objective value dropping sharply and stabilizing after around 40 generations. This indicates that the ENSGA-II algorithm can quickly explore promising regions in the solution space early on and then progressively approach the Pareto frontier through fine-tuned local searches in later stages, exhibiting a well-balanced mechanism between global exploration and local exploitation. In Figure 9(b), the convergence curve for time window violation time displays a significantly nonlinear decreasing trend, with the objective value dropping sharply within the first five generations and then converging to a lower layer. This performance suggests that the ENSGA-II algorithm can identify and avoid critical task scheduling conflicts early in the process. By adaptively adjusting crossover and mutation operations, it effectively maintains population diversity, continuously enhancing time coordination and improving overall scheduling punctuality. As shown in Figure 9(c), for the number of UAVs, the convergence curve exhibits a characteristic segmented decreasing trend, with alternating phases of “descent–plateau.” The initial descent phase reflects the algorithm’s ability to rapidly compress resource allocation through task aggregation and path integration strategies. The subsequent plateau phases indicate stabilization near local optima, which is beneficial for maintaining solution feasibility and diversity. As the optimization progresses, the algorithm transitions across multiple local plateaus, further reducing the scheduling scale. The final convergence result shows that despite a significant reduction in UAV numbers, task coverage remains effective, demonstrating that ENSGA-II achieves robust scheduling performance and optimization depth under multi-objective trade-offs.

In summary, the convergence processes of all three objectives validate the ENSGA-II algorithm’s stability and efficiency in handling high-dimensional, multi-constraint scheduling problems. The algorithm not only delivers rapid convergence and high-quality solution sets but also exhibits the ability to balance economic efficiency, timeliness, and resource feasibility, making it well-suited for UAV scheduling optimization in multi-depot scenarios.

4.3.2. Comparative Algorithm Analysis

To perform a comparative analysis, a standardized experimental environment was constructed. In this experiment, the number of customer points was set to 100, and the number of distribution centers was set to 3. The detailed parameters for each distribution center are shown in Table 6. Additionally, two types of heterogeneous UAVs were deployed collaboratively to execute tasks, with the parameters of each UAV type listed in Table 7. The customer task time window was set to the interval [0, 90], and the operating time for each distribution center was uniformly limited to 8 hours, comprehensively reflecting complex constraints related to both time and resources.

Table 6. Distribution center parameter settings.

Distribution center No.	Task coordinates	Time window constraint/h
J1	(200,200)	8
J2	(800,200)	8
J3	(500,800)	8

Table 7. UAV parameter settings.

Parameter name	Parameter value
Fixed cost/CNY	[500, 200]
Velocity/km·min ⁻¹	[20, 15]
Max distance/km	[2000, 1500]
Capacity /kg	[50, 150]
Variable cost /CNY·km ⁻¹	[2, 3]

Additionally, the task planning environment was constructed on a standardized grid map with dimensions of 1000×1000. This design not only ensures the spatial uniformity of task point distribution but also enhances the applicability and scalability of the model across various real-world scenarios. To evaluate the comprehensive optimization capability of the algorithms in terms of economic efficiency, timeliness, and resource utilization, five representative multi-objective evolutionary algorithms were selected for comparison: MOEA/D [41], NSGA-II [42], INSGA-II [43], PESA-II[44], and NSGA-III[45]. Three mainstream performance indicators were introduced to assess the optimization performance across the three objectives: the Hypervolume (HV), Inverted Generational Distance (IGD), and C-Metric (C-metric). In addition, to enhance the statistical significance and robustness of the results, each algorithm was independently executed 30 times, and the average performance over these runs was analyzed. All algorithms were tested under identical parameter settings to ensure consistency and fairness. The initial population size was set to 250, and the maximum number of generations was fixed at 100.

Table 8 summarizes the average performance of ENSGA-II and the five comparative algorithms across the three objectives, providing a comprehensive evaluation of their strengths in terms of solution set distribution, convergence behavior, and dominance ability.

Table 8. Performance comparison of different algorithms in the task planning problem.

Evaluation indicators	HV		IGD		C-metric	
ENSGA-II VS INSGA-II	ENSGA-II	INSGA-II	ENSGA-II	INSGA-II	ENSGA-II	INSGA-II
	1.07707541	1.02501847	0.02952997	0.09452646	0.63996333	0.12813667
ENSGA-II VS MOEAD	ENSGA-II	MOEAD	ENSGA-II	MOEAD	ENSGA-II	MOEAD
	1.07707541	0.54776679	0.02952997	0.28769867	1	0
ENSGA-II VS NSGA-II	ENSGA-II	NSGA-II	ENSGA-II	NSGA-II	ENSGA-II	NSGA-II
	1.07707541	0.59383683	0.02952997	0.21102712	0.97712	0
ENSGA-II VS NSGA-III	ENSGA-II	NSGA-III	ENSGA-II	NSGA-III	ENSGA-II	NSGA-III
	1.07707541	0.51902362	0.02952997	0.26770658	0.9802033	0
ENSGA-II VS PESA-II	ENSGA-II	PESA-II	ENSGA-II	PESA-II	ENSGA-II	PESA-II
	1.07707541	0.74747048	0.02952997	0.18581429	0.97266333	0.0027067
W/S/B	0/0/6		0/0/6		0/0/6	

In terms of the Hypervolume (HV) indicator, which reflects solution set diversity and coverage, ENSGA-II achieved an average value of 1.0771, significantly outperforming all benchmark algorithms. Specifically, it surpassed INSGA-II (1.0250), PESA-II (0.7475), NSGA-II (0.5938), MOEA/D (0.5478), and NSGA-III (0.5190) by approximately 5.08%, 44.06%, 81.33%, 96.52%, and 107.49%, respectively. This result indicates that ENSGA-II possesses superior capability in expanding the Pareto front in high-dimensional objective spaces. It effectively enhances the coverage and uniformity of the solution set, ensuring a well-distributed Pareto front across the global objective space. Regarding the IGD indicator, which measures convergence accuracy and proximity to the ideal Pareto front, ENSGA-II also demonstrated outstanding performance. Its average IGD value reached 0.0295, substantially lower than those of INSGA-II (0.0945), PESA-II (0.1858), NSGA-II (0.2110), NSGA-III (0.2677), and MOEA/D (0.2877). This indicates that the solutions generated by ENSGA-II are, on average, closest to the theoretical optimal Pareto front. The result underscores the algorithm’s ability to achieve faster and more stable convergence in complex nonlinear search spaces, significantly reducing deviation and enhancing the reliability and effectiveness of global solutions. For the C-metric indicator, which evaluates the dominance ratio of the Pareto front, ENSGA-II showed strong superiority in all pairwise comparisons with other algorithms. For instance, its C-metric value against MOEA/D was 1.0000,

indicating that all solutions generated by ENSGA-II dominated those from MOEA/D. In comparisons with NSGA-II and NSGA-III, the values reached 0.9771 and 0.9802, respectively, further illustrating ENSGA-II’s overwhelming advantage in terms of dominance. These results validate ENSGA-II’s leading performance in multi-objective optimization from the perspective of Pareto dominance, highlighting its significant global dominance capability. Finally, from the win-same-loss (W/S/B) statistical results, ENSGA-II emerged as the overall winner in all algorithmic comparisons, without any ties or inferior outcomes. This further confirms, at a statistical layer, the algorithm’s broad adaptability and robustness in solving complex scheduling problems. In summary, ENSGA-II achieved outstanding results across all evaluation metrics, demonstrating stronger global search capability and superior solution quality in high-dimensional multi-objective optimization tasks.

Table 9 further compares the performance of ENSGA-II and the five benchmark algorithms at the individual solution layer across the three objective functions. The comparison is based on pairwise evaluation of individual objective values, where the frequency of ENSGA-II outperforming (<), equaling (=), or underperforming (>) each benchmark algorithm is counted. This quantifies the dominance strength and optimization stability of ENSGA-II at the single-objective layer.

Table 9. Statistical comparison of different optimization algorithms.

Objective function Model comparison	f_1			f_2			f_3		
	<	=	>	<	=	>	<	=	>
ENSGA-II VS INSGA-II	11	1	18	30	0	0	7	18	5
ENSGA-II VS MOEAD	30	0	0	30	0	0	30	0	0
ENSGA-II VS NSGA-II	30	0	0	30	0	0	30	0	0
ENSGA-II VS NSGA-III	30	0	0	30	0	0	30	0	0
ENSGA-II VS PESA-II	30	0	0	30	0	0	30	0	0
Total	127	6	17	150	0	0	127	18	5

As shown in Table 9 , for the economic cost objective, ENSGA-II demonstrated absolute superiority in all 30 comparative runs against MOEA/D, NSGA-II, NSGA-III, and PESA-II (i.e., 30 out of 30 comparisons showed ENSGA-II as better, denoted by “<”, with “=” and “>” both being zero). However, in comparison with INSGA-II, ENSGA-II showed inferior performance in 18 instances and superior performance in only 11, indicating a certain disadvantage relative to INSGA-II in this single objective. This result may be attributed to INSGA-II’s improved strategy being more inclined toward local search optimization, which allows it to converge more rapidly to a locally minimal cost solution in some runs, especially in solution sets where cost minimization dominates. In contrast, ENSGA-II, as a structurally enhanced algorithm, is designed with a stronger emphasis on collaborative evolution across multiple objectives and the expansion of the overall Pareto frontier. Consequently, in striving for solution diversity and balance, it may sacrifice extreme optimality in individual objectives. Nonetheless, from a comprehensive multi-objective optimization perspective, ENSGA-II still shows a clear advantage in overall performance across all three objectives, demonstrating superior global scheduling capability and algorithmic stability. For the task delay time objective, ENSGA-II consistently outperformed all baseline algorithms, achieving 150 “<” outcomes without any ties (“=”) or inferior results (“>”), reflecting its strong capability in time-efficiency control. This confirms the algorithm’s effectiveness in avoiding time window violations and ensuring high layers of timeliness. Regarding the number of UAVs required, ENSGA-II also exhibited strong competitiveness. Out of 150 comparisons, it achieved 127 superior, 18 equal, and only 5 inferior results, fully demonstrating its excellence in UAV resource allocation efficiency. This indicates that ENSGA-II can effectively distribute tasks using fewer UAV platforms, thereby reducing resource waste.

In summary, across the three objectives, ENSGA-II achieved a win ratio of 90.7% (408 out of 450) across 90 experimental runs, with an inferior rate of only 5.1%, and showed an “all-win” dominance across multiple objectives. This not only confirms the algorithm’s global effectiveness in solution space exploration, but also highlights its practical applicability and robustness in addressing multi-

objective, resource-constrained, and highly constrained joint optimization problems, offering an efficient intelligent optimization tool for tackling high-dimensional complex scheduling challenges.

5. Conclusion

This study addresses the challenges of modeling complexity, high-dimensional coupling, and multi-objective collaborative optimization in the multi-UAV and multi-depot task planning problem. A bi-layer optimization framework with an upper-lower coordination mechanism is developed, along with improved algorithms tailored to different hierarchical optimization needs. The objective is to enhance the adaptability and optimization capability of task planning systems in complex urban airspace logistics environments. The main contributions of this study are summarized as follows:

(1) To tackle the modeling complexity and multi-objective optimization challenges of the multi-UAV and multi-depot task planning problem, a novel bi-layer task planning model is proposed. The upper-layer model addresses task allocation, while the lower-layer model focuses on path planning. Information such as task assignment schemes and flight distances is exchanged between layers through an interaction and feedback mechanism, thereby improving the model's solving efficiency in multi-objective, multi-constraint environments.

(2) For the task allocation problem, an ENSGA-II algorithm is proposed. It incorporates a heuristic initialization strategy under path feasibility constraints, goal-oriented top-down and bottom-up search operators, and an adaptive mutation mechanism. This design enables dynamic coordination between global exploration and local exploitation, providing robust and efficient optimization support for upper-layer task assignment.

(3) For the path planning problem, an improved PSO algorithm is proposed by integrating chaotic initialization, linearly adjusted acceleration factors and maximum velocity, particle position updates with stochastic perturbation, and adaptively tuned inertia weights. These improvements enhance the solving capacity and quality of path generation.

In the simulation experiments, the proposed bi-layer multi-objective collaborative scheduling optimization model and ENSGA-II algorithm demonstrated excellent performance on benchmark instances of the multi-depot-multi-UAV task planning problem. In terms of task allocation metrics, the optimization results show an average reduction of 47.8% in economic cost, a 71.4% decrease in the number of UAVs required, and significant compression of time window violation durations. Regarding algorithmic performance, the proposed ENSGA-II algorithm achieved an HV value of 1.0771, outperforming other algorithms by 72.35% to 109.82%. Its average IGD value was 0.0295, far superior to other baseline algorithms (ranging from 0.0893 to 0.2714), and it exhibited overwhelming dominance in the C-metric, indicating significant advantages in the distribution, convergence, and diversity of non-dominated solutions. These results validate the applicability of the proposed bi-layer model and the robustness and efficiency of the designed algorithm in complex scheduling scenarios.

The bi-layer coordinated optimization model and the improved algorithms developed in this study provide an effective solution for dealing with complex constraints, conflicting objectives, and large-scale task challenges. They offer a highly adaptive and high-quality decision-making tool for collaborative task planning in urban airspace logistics scenarios. Future work could extend the current model to more complex dynamic environments by considering real-time task changes, complex airspace constraints, and various uncertainties, thereby enhancing the model's adaptability and robustness. Furthermore, how to apply this model to real-world UAV scheduling systems remains a promising direction for future research.

References

1. Darchini-Tabrizi, M.; Pakdaman-Donyavi, A.; Entezari-Maleki, R.; Sousa, L. Performance enhancement of UAV-enabled MEC systems through intelligent task offloading and resource allocation. *Computer Networks* **2025**, *264*, 111280. [\[CrossRef\]](#)
2. Kurdi, H.A.; Aloboud, E.; Alalwan, M.; Alhassan, S.; Alotaibi, E.; Bautista, G.; How, J.P. Autonomous task allocation for multi-UAV systems based on the locust elastic behavior. *Applied Soft Computing* **2018**, *71*, 110–126. [\[CrossRef\]](#)
3. 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. *Computer Communications* **2023**, *207*, 46–55. [\[CrossRef\]](#)
4. Zhang, D.; Zhang, Z.; Zhang, J.; Zhang, T.; Zhang, L.; Chen, H. UAV-assisted task offloading system using dung beetle optimization algorithm & deep reinforcement learning. *Ad Hoc Networks* **2024**, *156*, 103434. [\[CrossRef\]](#)
5. Zhang, C.; Guo, J.; Wang, F.; Chen, B.; Fan, C.; Yu, L.; Wang, Z. A dynamic parameters genetic algorithm for collaborative strike task allocation of unmanned aerial vehicle clusters towards heterogeneous targets. *Applied Soft Computing* **2025**, *175*, 113075. [\[CrossRef\]](#)
6. Wang, X.; Gao, X.; Wang, L.; Su, X.; Jin, J.; Liu, X.; Deng, Z. Resilient multi-objective mission planning for UAV formation: A unified framework integrating task pre- and re-assignment. *Defence Technology* **2025**, *45*, 203–226. [\[CrossRef\]](#)
7. He, C.; Ouyang, H.; Huang, W.; Li, S.; Zhang, C.; Ding, W.; Zhan, Z.-H. An adaptive heuristic algorithm with a collaborative search framework for multi-UAV inspection planning. *Applied Soft Computing* **2025**, *174*, 112969. [\[CrossRef\]](#)
8. Ahmed, G.; Sheltami, T.; Ghaleb, M.; Hamdan, M.; Mahmoud, A.; Yasar, A. Energy-Efficient Internet of Drones Path-Planning Study Using Meta-Heuristic Algorithms. *Applied Sciences* **2024**, *14*(6), 2418. [\[CrossRef\]](#)
9. Li, K.; Yan, X.; Han, Y. Multi-mechanism swarm optimization for multi-UAV task assignment and path planning in transmission line inspection under multi-wind field. *Applied Soft Computing* **2024**, *150*, 111033. [\[CrossRef\]](#)
10. Xiao, Y.; Li, Y.; Liu, H.; Chen, Y.; Wang, Y.; Wu, G. Adaptive large neighborhood search algorithm with reinforcement search strategy for solving extended cooperative multi task assignment problem of UAVs. *Information Sciences* **2024**, *679*, 121068. [\[CrossRef\]](#)
11. Fan, M.; Liu, H.; Wu, G.; Gunawan, A.; Sartoretti, G. Multi-UAV reconnaissance mission planning via deep reinforcement learning with simulated annealing. *Swarm and Evolutionary Computation* **2025**, *93*, 101858. [\[CrossRef\]](#)
12. Li, H.; Huang, J. Hierarchical federated deep reinforcement learning based joint communication and computation for UAV situation awareness. *Vehicular Communications* **2024**, *50*, 100853. [\[CrossRef\]](#)
13. Wu, Y.; Gou, J.; Ji, H.; Deng, J. Hierarchical mission replanning for multiple UAV formations performing tasks in dynamic situation. *Computer Communications* **2023**, *200*, 132–148. [\[CrossRef\]](#)
14. Liu, Z.; Li, Y.; Wu, Y. Multiple UAV formations delivery task planning based on a distributed adaptive algorithm. *Journal of the Franklin Institute* **2023**, *360*(4), 3047–3076. [\[CrossRef\]](#)
15. Cheng, Z.; Zhao, L.; Shi, Z. Decentralized Multi-UAV Path Planning Based on Two-Layer Coordinative Framework for Formation Rendezvous. *IEEE Access* **2022**, *10*, 45695–45708. [\[CrossRef\]](#)
16. Yan, F.; Chu, J.; Hu, J.; Zhu, X. Cooperative task allocation with simultaneous arrival and resource constraint for multi-UAV using a genetic algorithm. *Expert Systems with Applications* **2024**, *245*, 123023. [\[CrossRef\]](#)
17. Zhan, H.; Zhang, Y.; Huang, J.; et al. A reinforcement learning-based evolutionary algorithm for the unmanned aerial vehicles maritime search and rescue path planning problem considering multiple rescue centers. *Memetic Computing* **2024**, *16*, 373–386. [\[CrossRef\]](#)
18. Xue, K.; Zhai, L.; Li, Y.; Lu, Z.; Zhou, W. Task offloading and multi-cache placement based on DRL in UAV-assisted MEC networks. *Vehicular Communications* **2025**, *53*, 100900. [\[CrossRef\]](#)
19. Shianios, D.; Kolios, P.; Kyrkou, C. MultiFire20K: A semi-supervised enhanced large-scale UAV-based benchmark for advancing multi-task learning in fire monitoring. *Computer Vision and Image Understanding* **2025**, *254*, 104318. [\[CrossRef\]](#)
20. Yildiz, B.; Aslan, M.F.; Durdu, A.; Kayabasi, A. Consensus-based virtual leader tracking swarm algorithm with GDRRT*-PSO for path-planning of multiple-UAVs. *Swarm and Evolutionary Computation* **2024**, *88*, 101612. [\[CrossRef\]](#)

21. Wang, J.; Wu, Y.X.; Chen, Y.-Q.; Ju, S. Multi-UAVs collaborative tracking of moving target with maximized visibility in urban environment. *Journal of the Franklin Institute* **2022**, 359(11), 5512–5532. [\[CrossRef\]](#)
22. Hu, G.; Cheng, M.; Houssein, E.H.; Jia, H. CMPSO: A novel co-evolutionary multigroup particle swarm optimization for multi-mission UAVs path planning. *Advanced Engineering Informatics* **2025**, 63, 102923. [\[CrossRef\]](#)
23. Li, Y.; Zhang, Z.; Sun, Q.; Huang, Y. An improved ant colony algorithm for multiple unmanned aerial vehicles route planning. *Journal of the Franklin Institute* **2024**, 361(15), 107060. [\[CrossRef\]](#)
24. Huang, S.; Chen, W.; Lu, B.; Xiao, F.; Shen, C.; Zhang, W. An improved BAT algorithm for collaborative dynamic target tracking and path planning of multiple UAV. *Computers and Electrical Engineering* **2024**, 118, 109340. [\[CrossRef\]](#)
25. Shen, Q.; Zhang, D.; He, Q.; Ban, Y.; Zuo, F. A novel multi-objective dung beetle optimizer for Multi-UAV cooperative path planning. *Heliyon* **2024**, 10(17), e37286. [\[CrossRef\]](#)
26. Heidari, A.; Orazani, S.M.H.; Khalilzadeh, M.; Jolai, F. A multi-objective model for cooperative delivery of customer orders using multiple trucks and UAVs considering weather conditions. *Internet of Things (The Netherlands)* **2025**, 29, 101468. [\[CrossRef\]](#)
27. Bi, W.; Zhang, M.; Chen, H.; Zhang, A. Cooperative task allocation method for air-sea heterogeneous unmanned system with an application to ocean environment information monitoring. *Ocean Engineering* **2024**, 309, 118496. [\[CrossRef\]](#)
28. Akay, R.; Yildirim, M.Y. SBA*: An efficient method for 3D path planning of unmanned vehicles. *Mathematics and Computers in Simulation* **2025**, 231, 294–317. [\[CrossRef\]](#)
29. Huang, J.; Chen, C.; Shen, J.; Liu, G.; Xu, F. A self-adaptive neighborhood search A-star algorithm for mobile robots global path planning. *Computers and Electrical Engineering* **2025**, 123, 110018. [\[CrossRef\]](#)
30. Yildirim, M.Y.; Akay, R. An efficient grid-based path planning approach using improved artificial bee colony algorithm. *Knowledge-Based Systems* **2025**, 318, 113528. [\[CrossRef\]](#)
31. Ruan, F.; Chen, C.; He, C.; Cheng, Y.; Sun, Y. Optimization method of public decontamination location and allocation problem in off-site nuclear emergency based improved NSGA-II. *Journal of Hazardous Materials* **2025**, 489, 137572. [\[CrossRef\]](#)
32. Yin, Q.; Wu, G.; Sun, G.; Gu, Y. Multi-objective orbital maneuver optimization of multi-satellite using an adaptive feedback learning NSGA-II. *Swarm and Evolutionary Computation* **2025**, 93, 101835. [\[CrossRef\]](#)
33. He, R.; Bian, R.; Hua, J.; Zhao, L.; Xu, F.; Long, J. Multi-objective optimization of gasoline blending scheduling via NSGA-II algorithm with composite operators considering oil movement path planning. *Expert Systems with Applications* **2025**, 280, 127426. [\[CrossRef\]](#)
34. Srivastava, G.; Singh, A.; Mallipeddi, R. NSGA-II with objective-specific variation operators for multiobjective vehicle routing problem with time windows. *Expert Systems with Applications* **2021**, 176, 114779. [\[CrossRef\]](#)
35. Xu, W.; Cheng, L.; Cui, Y.; Yan, H.; Jiang, H.; Jiao, W. An improved method of applying PSO-BP to optimize PAT energy efficiency based on entropy production theory. *Energy Conversion and Management* **2025**, 327, 119472. [\[CrossRef\]](#)
36. Xiang, X.; Yan, X.; Gao, C.; Zhu, S.; Xi, M.; Gao, H. A circle chaos random search strategy particle swarm optimization with its application. *Computers and Electrical Engineering* **2022**, 102, 108219. [\[CrossRef\]](#)
37. Ibrahim, A.-W.; Xu, J.; Al-Shamma'a, A.A.; Farh, H.M.H.; Dagal, I. Intelligent adaptive PSO and linear active disturbance rejection control: A novel reinitialization strategy for partially shaded photovoltaic-powered battery charging. *Computers and Electrical Engineering* **2025**, 123, 110037. [\[CrossRef\]](#)
38. Wang, F.; Yang, Q. Two-layer task planning method for multi-UAV logistics distribution. *Journal of Beijing University of Aeronautics and Astronautics* **2023**. [\[CrossRef\]](#)
39. Tatsumi, K.; Ibuki, T.; Tanino, T. Particle swarm optimization with stochastic selection of perturbation-based chaotic updating system. *Applied Mathematics and Computation* **2015**, 269, 904–929. [\[CrossRef\]](#)
40. Zhao, J.; Cui, Y.; Huang, J.; Zhu, R. Adaptive grey wolf optimizer based on transfer function inertia weight of second-order high-pass filter. *Alexandria Engineering Journal* **2024**, 107, 443–454. [\[CrossRef\]](#)
41. Chen, X.; Luo, D.; Yu, D.; Fang, Z. Multi-objective test case prioritization based on an improved MOEA/D algorithm. *Expert Systems with Applications* **2025**, 266, 126086. [\[CrossRef\]](#)
42. Wen, X.; Zhang, X.; Li, H.; Ji, S.; Wang, H.; Ye, G.; Xing, H.; Liu, S. An improved NSGA-II algorithm based on reinforcement learning for aircraft moving assembly line integration optimization problem. *Swarm and Evolutionary Computation* **2025**, 94, 101911. [\[CrossRef\]](#)

43. Srivastava, G.; Singh, A.; Mallipeddi, R. NSGA-II with objective-specific variation operators for multiobjective vehicle routing problem with time windows. *Expert Systems with Applications* **2021**, *176*, 114779. [\[CrossRef\]](#)
44. Khaleghi, A.; Eydi, A. Multi-period hub location problem considering polynomial time-dependent demand. *Computers & Operations Research* **2023**, *159*, 106357. [\[CrossRef\]](#)
45. Hou, Y.; Liao, X.; Chen, G.; Chen, Y. Co-Evolutionary NSGA-III with deep reinforcement learning for multi-objective distributed flexible job shop scheduling. *Computers & Industrial Engineering* **2025**, *203*, 110990. [\[CrossRef\]](#)

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