drones

MDPI

Article

A Risk-Averse Two-Stage Stochastic Programming Model for Emergency UAV Task Allocation

Shumeng Xu, Lili Wan, Jiahui Huang, Qingyang Zhang, Zhenyu Yuan and Zhan Wang *

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; xushumeng@nuaa.edu.cn (S.X.); wanlili@nuaa.edu.cn (L.W.); hjhhjh@nuaa.edu.cn (J.H.); zhangqingyang@nuaa.edu.cn (Q.Z.); xiaoyu82@nuaa.edu.cn (Z.Y.)
* Correspondence: wangzhan@nuaa.edu.cn

Highlights

What are the main findings?

- A Mean-CVaR-based two-stage stochastic programming model is proposed to extend emergency UAV task allocation from expected-cost scheduling to risk-averse decision-making that accounts for both average performance and tail delay risk.
- A problem-tailored Enhanced BD framework integrating Partial Benders, heuristic warm start, and dynamic cut-pool management is developed for the proposed multi-scenario risk-averse model.

What are the implications of the main findings?

- Across ten independent replications, Enhanced BD achieves a higher convergence success rate and lower final BD Gap than Basic BD in the medium-sized and largest tested instances.
- The framework provides a feasible way to identify tail delay risk in emergency UAV task chains and can inform UAV task allocation and tail delay risk assessment under uncertainty.

Abstract

As UAVs are increasingly used in emergency rescue, task allocation under uncertainty still faces tail delay risk. Existing studies mainly optimize expected cost and pay insufficient attention to task temporal relations and delay losses under extreme scenarios. To address this issue, this study develops a risk-averse two-stage stochastic programming model that incorporates the precedence relation between reconnaissance and delivery tasks, UAV routes, and task execution sequences into a unified decision process. The first stage determines task assignment, route selection, and visit order, while the second stage evaluates waiting, delay, and recourse costs under stochastic scenarios. A Mean-CVaR risk measure is adopted to characterize both average performance and tail risk. To solve the resulting multi-scenario risk-averse model, this study develops a problem-tailored Enhanced BD framework based on the classical Benders decomposition structure. The proposed framework integrates partial scenario embedding, heuristic warm start, and dynamic cut-pool management to strengthen early master problem information, improve feasible-route search, and control the growth of scenario-wise cuts. Numerical experiments based on a Nanjing emergency rescue instance evaluate the model and algorithm in terms of solution performance, acceleration ablation, optimized scheduling results, and parameter sensitivity. The results show that the proposed model can identify tail delay risk concentrated at a small number of demand points and downstream nodes in task chains. Across ten independent replications,

Check for updates

Academic Editor: Paraskevi Zacharia

Received: 20 May 2026
Revised: 5 July 2026
Accepted: 11 July 2026
Published: 12 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Enhanced BD achieves a higher convergence success rate and lower final BD Gap than Basic BD in the medium-sized and largest tested instances. Parameter analysis shows that moderate risk aversion improves out-of-sample performance, whereas excessive risk aversion or resource allocation may reduce overall scheduling efficiency. The proposed method improves tail risk identification and solution capability for emergency UAV task allocation under time uncertainty and provides a methodological reference for risk-aware UAV emergency scheduling.

Keywords: emergency UAV task allocation; two-stage stochastic programming; Mean-CVaR; risk aversion; Benders decomposition; tail delay risk

1. Introduction

The increasing frequency of natural disasters, emergencies, and public safety incidents has raised the demand for faster and more efficient emergency response [1]. When roads are blocked, information is incomplete, and demand evolves over time, conventional ground-based rescue operations often face limited accessibility, response delays, and delayed information feedback. Unmanned aerial vehicles (UAVs), which are less constrained by terrain and can be deployed rapidly, have therefore been increasingly used for disaster reconnaissance, emergency-material delivery, and related tasks. As low-altitude emergency logistics and rescue technologies develop, effective UAV fleet coordination and response efficiency have become important research issues [2].

Emergency UAV task allocation has therefore received growing attention [2]. In rescue operations, multiple task types such as reconnaissance, assessment, and delivery often exhibit explicit precedence relations. For example, some delivery tasks can be executed only after the corresponding reconnaissance tasks have been completed; therefore, task assignment, routing, and execution sequence need to be determined jointly. Existing studies often use time windows, task sequences, or precedence constraints to ensure scheduling feasibility, but they provide limited analysis of how changes in predecessor-task completion times affect downstream tasks [3]. Meanwhile, weather conditions, UAV payload, and on-site operations may cause fluctuations in flight time and service time. Although existing studies have considered uncertain factors such as flight time and task demand [4], these disturbances are often modeled at the level of individual operations or overall routes, and their propagation through reconnaissance–delivery precedence relations into waiting and delay remains insufficiently characterized. Emergency UAV task allocation naturally involves pre-event decisions and scenario-dependent responses. Two-stage stochastic programming can separate task assignment and routing decisions before uncertainty is revealed from time responses after uncertainty is realized, making it suitable for this problem [5].

Existing UAV task allocation studies under uncertainty often focus on expected cost, robust feasibility, or average completion performance, with limited attention to tail delay losses under a small number of high-disturbance scenarios. In task chains with precedence relationships, however, a few high-disturbance scenarios may lead to severe delays in critical delivery tasks. Focusing only on average performance may underestimate such tail losses and cannot fully reflect decision-makers' risk preferences [6]. To address this issue, this study adopts the Mean-CVaR criterion, which combines average cost with tail loss. In the objective function, the expectation term evaluates average performance under ordinary scenarios, whereas the CVaR term measures the average loss exceeding a risk threshold at a given confidence level [7]. Mean-CVaR balances average performance and extreme losses

and is suitable for uncertain decision problems with different risk preferences [8]. For the case in which reconnaissance task delays postpone the start of the corresponding delivery tasks and further affect downstream nodes, this study explicitly models the constraint from reconnaissance completion time to delivery start time in a two-stage stochastic program, computes waiting and delay caused by stochastic flight and service times in each scenario, and uses Mean-CVaR to evaluate downstream task-chain risk in high-loss scenarios.

When discrete routing decisions, multi-scenario evaluation, and a risk-averse objective are incorporated simultaneously, the deterministic equivalent model grows rapidly [9], and existing methods often have limited solution capability [10]. To solve the proposed risk-averse two-stage stochastic programming model, this study develops a problem-tailored Enhanced BD framework based on the classical Benders decomposition structure. Benders decomposition separates the original model into a master problem and scenario subproblems and solves them iteratively, thereby reducing the difficulty of directly solving the deterministic equivalent model. The proposed framework further combines partial scenario embedding, heuristic warm start, and dynamic cut-pool management to strengthen early master problem information, improve feasible-route search, and control the growth of scenario-wise cuts.

The main contributions of this study are as follows: (1) A two-stage stochastic optimization model is developed for emergency UAV task allocation. reconnaissance–delivery precedence relationships are modeled together with task assignment, routing, and execution sequence, and the effects of stochastic flight and service times on waiting and delay are characterized. (2) A Mean-CVaR risk-averse objective is introduced to jointly characterize average operational performance and tail risk. (3) A problem-tailored Enhanced BD computational framework is developed by integrating partial scenario embedding, heuristic warm start, and dynamic cut-pool management. The individual and combined computational effects of these components are evaluated through cross-scale ablation experiments. (4) Numerical experiments analyze the effects of training sample size, fleet size, risk parameters, uncertainty intensity, network topology, and tested instance size on solution quality and algorithmic performance, thereby evaluating the applicability of risk-averse modeling and the effectiveness of the proposed framework.

The remainder of this paper is organized as follows. Section 2 reviews related studies on emergency UAV task allocation, risk-averse modeling, and Benders decomposition. Section 3 presents the two-stage stochastic programming model and problem description. Section 4 introduces Mean-CVaR risk-averse modeling and its linear reformulation. Section 5 presents the problem-tailored Enhanced BD solution framework and its acceleration strategies. Section 6 reports numerical results and performance analyses. Finally, Section 7 concludes the paper and discusses future research directions.

2. Related Work

2.1. Emergency UAV Task Allocation

Search-and-rescue tasks in natural disasters, emergencies, and public safety incidents are typically time-sensitive, environment-dependent, and constrained by limited ground accessibility. Traditional ground-based rescue forces often cannot cover all task areas in a timely manner. UAVs offer rapid deployment, flexible mobility, and low-altitude sensing capabilities, making them important auxiliary tools for search-and-rescue operations and extending their use to post-disaster inspection, material delivery, medical support, and on-site monitoring [11].

Existing studies have mainly examined emergency UAV scheduling from the perspectives of task assignment and routing optimization. Tang et al. [12] studied multi-UAV task planning in post-disaster emergency logistics and developed a task scheduling method that

minimizes task completion time and earliness/tardiness penalty costs. Rottondi et al. [13] incorporated emergency services such as monitoring, material delivery, and communication support into a unified scheduling framework for multi-service UAV operations in post-disaster scenarios. Chowdhury et al. [14] developed a routing optimization model for a heterogeneous fixed UAV fleet in post-disaster inspection. Lin et al. [15] studied coordinated task assignment and routing optimization for a heterogeneous UAV fleet in emergency healthcare logistics under constraints such as material supply, time windows, and geographical conditions. These studies provide a basis for emergency UAV task allocation, but most assume relatively certain task execution conditions and give limited consideration to execution-stage disturbances.

Post-disaster rescue environments are highly uncertain, and deterministic models cannot adequately capture the effects of environmental disturbances on scheduling decisions. Accordingly, recent studies have incorporated uncertainty into emergency task allocation models. Gao et al. [16] considered uncertain flight fuel consumption in a cooperative task allocation problem for heterogeneous UAVs with time-window constraints and transformed the model into a deterministic equivalent form through adjustable robust optimization. Huang et al. [17] addressed dynamic task allocation for heterogeneous multi-UAV systems in uncertain environments. Based on binary interval number theory, they constructed a fuzzy multi-constraint programming model that incorporates target-position information deviation, uncertain task execution time, and UAV survival probability, and solved it using dynamic time slices and the 4DI-GWO algorithm. Ghelichi et al. [18] introduced demand uncertainty into drone logistics for disaster-impacted populations and analyzed the effect of post-disaster demand fluctuations on delivery decisions. Yin et al. [4] considered both demand uncertainty and ground travel-time fluctuation in a robust vehicle routing problem with drones for humanitarian logistics. Although these studies incorporate uncertainty into task allocation and routing decisions, most focus on demand, energy consumption, or travel-time disturbances. Time fluctuations during UAV task execution and the resulting delay propagation remain less explored.

In addition to random disturbances, emergency tasks often involve time windows, execution sequences, and precedence relationships. Time windows define the feasible execution interval of an individual task, execution sequences describe the visiting order of tasks along UAV routes, and precedence relationships require a successor task to start only after a specified predecessor task has been completed, thereby creating explicit temporal dependencies among tasks. Existing multi-robot task allocation studies show that tasks with precedence relationships cannot be determined solely by task assignment; task allocation and execution scheduling must be considered jointly. Nunes et al. [19] studied multi-robot task allocation with temporal and precedence constraints and showed that temporal dependencies among tasks affect the feasibility of allocation plans. Ferreira et al. [20] further incorporated cross-schedule dependencies, precedence relationships, and temporal scheduling into a unified task allocation and scheduling process for heterogeneous multi-robot teams. In multi-UAV task allocation, Cui et al. [21] proposed a distributed task allocation method for multi-UAV systems with complex time-window constraints, enabling conflict-free task assignments within feasible time windows. Wang et al. [22] studied heterogeneous UAV task planning with temporal constraints by treating flight time as a random variable and analyzing its effect on task start times, completion-time distributions, and solution robustness. Jeong et al. [23] investigated task allocation with temporal constraints in heterogeneous multi-UAV systems, aiming to minimize makespan and using waiting time to handle temporal relations among tasks. Liu et al. [3] incorporated temporal task chains into heterogeneous UAV swarm task allocation to capture precedence dependencies among multi-stage tasks such as disaster reconnaissance, material deliv-

ery, and effect assessment. These studies show that time windows, task sequences, and precedence relationships jointly affect the feasibility and execution efficiency of UAV task allocation plans, providing a basis for unified scheduling of multi-stage emergency tasks such as reconnaissance and delivery.

However, existing studies mainly focus on the effects of deterministic time windows, task sequences, or precedence relationships on scheduling feasibility, while paying limited attention to how stochastic flight and service times affect downstream tasks through precedence relationships. For reconnaissance–delivery emergency tasks, the completion time of a predecessor reconnaissance task directly determines when the corresponding delivery task can start. When flight time or service time is disturbed, delays may propagate along subsequent task chains and lead to accumulated waiting and delay.

Based on this observation, this study investigates reconnaissance–delivery UAV task allocation under execution time uncertainty. Task assignment, route selection, execution sequence, and reconnaissance–delivery precedence relationships are incorporated into a unified decision process, and the effects of time disturbances on waiting, delay, and tail losses are further evaluated under stochastic scenarios.

2.2. Two-Stage Stochastic Optimization and Mean-CVaR Risk-Averse Modeling

To address uncertain factors such as demand, energy consumption, and fuel consumption in emergency scenarios, existing studies have applied stochastic programming, fuzzy programming, and robust optimization to UAV task allocation. Among these methods, two-stage stochastic programming can model uncertainty with known probability distributions and jointly describe advance decisions and scenario-dependent responses. It has therefore been used for uncertain decision modeling in UAV operations and emergency logistics. Dukkanci et al. [24] developed a two-stage stochastic programming model for UAV-based emergency delivery under uncertain post-earthquake demand and road-network conditions. He et al. [25] applied two-stage stochastic programming to UAV scheduling under uncertain energy consumption. Venkatachalam et al. [26] adopted a two-stage framework for multi-UAV routing under stochastic fuel consumption. Yan et al. [27] constructed a two-stage UAV routing model with time windows and rescheduling mechanisms under uncertain delivery reliability. These studies show that the two-stage framework can handle random demand, energy consumption, and reliability. For reconnaissance–delivery UAV tasks, however, the recourse stage also needs to describe waiting and delay costs induced by flight time, service time, and precedence relations.

For emergency tasks with strict timeliness and reliability requirements, average cost alone may fail to reflect losses under extreme scenarios. Rockafellar and Uryasev [7] proposed a computable formulation of conditional value-at-risk (CVaR). Fábián [28] systematically discussed CVaR objectives and constraints in two-stage stochastic models, providing a basis for embedding risk measures into stochastic programming. Miller and Ruszczyński [6] and Noyan [8] introduced the Mean-CVaR criterion into two-stage stochastic programming to balance average cost and tail risk. The former analyzed model structure and decomposition for risk-averse two-stage stochastic linear programming, while the latter applied Mean-CVaR to uncertain decision-making in disaster management. These studies indicate that Mean-CVaR can characterize tail risk within a two-stage stochastic programming framework. Subsequent studies further extended risk-averse modeling to more complex uncertain decision settings. Ling et al. [29] studied robust two-stage stochastic optimization with risk aversion. Wang et al. [30] developed a two-stage distributionally robust model based on the worst case Mean-CVaR criterion. Gu et al. [31] discussed a data-driven distributionally robust risk-averse model based on a Wasserstein ball. Ghaffarinasab et al. [32] applied Mean-CVaR to a risk-averse hub location problem.

Khatami et al. [33] further examined the modeling and analysis of related risk measures from the perspective of multistage stochastic programming.

Existing studies have examined uncertain decision-making from the perspectives of two-stage stochastic optimization and risk-averse modeling. In emergency UAV task allocation, however, average cost remains the main evaluation criterion in much of the literature, and tail delay risk under extreme scenarios is less fully considered. Accordingly, this study introduces the Mean-CVaR risk measure into a two-stage stochastic programming model to jointly evaluate the average performance and tail risk of UAV task allocation plans.

2.3. Benders Decomposition

As the numbers of tasks and scenarios increase, the deterministic equivalent formulation of a risk-averse two-stage stochastic programming problem expands rapidly, making direct solution increasingly difficult [9]. Benders decomposition provides a classical framework for addressing such problems by separating first-stage decisions from scenario-dependent recourse problems. Benders [34] introduced the fundamental idea of iteratively solving a master problem and a set of subproblems. Birge and Louveaux [35] subsequently developed the multicut approach for two-stage stochastic linear programming to exploit scenario-specific information, while Louveaux and Laporte [36] extended the framework to problems with integer first-stage variables, thereby laying the foundation for the L-shaped method in stochastic integer programming. Together, these studies established the basic methodology for decomposing integer decisions and scenario-based recourse structures in two-stage stochastic programs.

Building on this foundation, subsequent studies integrated Benders decomposition with CVaR or Mean-CVaR risk measures. Noyan [8] proposed single-cut and multicut L-shaped algorithms for risk-averse two-stage stochastic programming and demonstrated that a Mean-CVaR objective can be incorporated into the classical decomposition framework. Huang et al. [37] developed an improved Benders decomposition method for a two-stage unit commitment problem with CVaR constraints. Taghizadeh and Venkatachalam [38] investigated a risk-averse stochastic ordering problem with a CVaR-minimization objective. Ghaffarinasab et al. [32] combined Mean-CVaR, Benders decomposition, and scenario grouping for hub location, whereas Zhang et al. [39] designed a multicut Benders-based branch-and-cut method for a risk-averse gate assignment problem. These studies show that Benders decomposition can effectively accommodate the combination of risk measures and two-stage recourse structures. However, existing applications have mainly focused on energy scheduling, inventory control, facility location, and gate assignment, with relatively limited attention paid to UAV routing problems involving precedence-constrained task chains and scenario-dependent time propagation.

As problem size increases, researchers further improve the computational performance of Benders decomposition through enhanced cut generation, master problem strengthening, and scenario-processing strategies. Magnanti and Wong [40] introduced Pareto-optimal cuts to improve cut quality, while Rahmaniani et al. [10] provided a systematic review of acceleration techniques, including multicuts, stabilization, master problem enhancement, warm starts, and cut management. Crainic et al. [41] strengthened the master problem by embedding a subset of scenarios, Mahéo et al. [42] integrated branch-and-cut with Benders decomposition, and Ramírez-Pico et al. [43] proposed an adaptive cutting-plane method. More recently, Baena et al. [44], Zhang et al. [45], and Heitsch and Römisch [46] further improved the solution performance of large-scale stochastic programs through stabilization, adaptive oracles, and scenario reduction, respectively. Related techniques have also been extended to distributionally robust optimization and risk-averse transportation recovery problems [47,48].

Overall, existing research has established a relatively comprehensive methodological framework ranging from basic decomposition schemes and risk-averse extensions to various acceleration mechanisms. Nevertheless, individual strategies are typically designed to address a specific computational bottleneck. In the problem considered in this study, first-stage integer routing decisions must be repeatedly evaluated through scenario-specific time propagation and delay recourse, while the Mean-CVaR objective introduces additional tail-excess-loss variables. As a result, standard Benders decomposition may suffer from insufficient information in the master problem, slow identification of high-quality initial feasible solutions, and continuous growth in the number of generated cuts. To address these computational challenges simultaneously, this study integrates partial scenario embedding, a heuristic warm start, and dynamic cut-pool management into the Benders framework to construct a problem-tailored solution method for the proposed model.

2.4. Research Gap and Positioning of This Study

To clarify the research gap more systematically, Table 1 compares representative studies across four dimensions: uncertainty type, risk criterion, task temporal structure, and solution method, and further summarizes their limitations relative to the research problem addressed in this study.

Table 1. Systematic comparison of related studies and this study.

Study	Uncertainty Type	Risk Criterion	Task Temporal Structure	Solution Method	Comparative Role and Limitation
Lin et al. [15]	Not modeled	Cost/service efficiency	Time windows and service constraints	Task assignment and routing optimization	Represents emergency UAV task assignment and routing studies, but does not model stochastic execution time disturbances or tail delay risk
Liu et al. [3]	Not modeled	Deterministic task allocation objective	Task precedence and temporal task chains	Task allocation algorithm	Represents studies that consider task temporal relations, but mainly for task dependency and feasibility, without analyzing stochastic delay propagation or Mean-CVaR tail risk
Wang et al. [22]	Flight time	Robustness/ completion-time distribution	Temporal constraints	Task-planning model	Represents the combination of flight-time uncertainty and temporal constraints, but does not evaluate tail delay loss through a risk-averse recourse objective
Gao et al. [16]	Fuel consumption	Robust objective	Time-window constraints	Adjustable robust optimization	Represents robust uncertainty modeling in UAV task assignment, but does not model waiting and delay recourse under reconnaissance–delivery precedence
Dukkanci et al. [24]	Demand, road network	Expected cost	Not focused on precedence chains	Two-stage stochastic programming	Represents two-stage stochastic modeling in UAV relief logistics, but does not address reconnaissance–delivery precedence or chain-based delay recourse
Yan et al. [27]	Delivery reliability	Expected/ rescheduling objective	Time-window constraints	Two-stage routing optimization	Represents two-stage UAV routing with time windows and random reliability, but does not use a tail risk criterion to evaluate extreme delay losses
Miller and Ruszczyński [6]; Noyan [8]	Generic stochastic parameters	Mean-CVaR/CVaR	Non-UAV task-chain setting	L-shaped decomposition	Provides the basis for Mean-CVaR risk-averse stochastic programming, but does not address delay propagation in UAV task chains

Table 1. *Cont.*

Study	Uncertainty Type	Risk Criterion	Task Temporal Structure	Solution Method	Comparative Role and Limitation
Zhang et al. [39]	Arrival time	Risk-averse two-stage objective	Non-UAV task-chain setting	Multicut Benders	Represents the integration of risk-averse two-stage modeling and decomposition algorithms, but is not designed for UAV routing with reconnaissance–delivery precedence and scenario-dependent timing recourse
This study	Flight time, service time	Mean-CVaR	reconnaissance–delivery precedence chains	Enhanced BD	Models waiting, delay recourse, and tail delay risk caused by execution time disturbances in precedence-constrained task chains, and constructs a solution framework tailored to this structure

As shown in Table 1, existing studies have separately considered execution time uncertainty, task temporal relationships, and risk-averse decision-making, but their combined effects in emergency UAV task allocation remain insufficiently explored. Time windows and precedence relationships are often used mainly to ensure scheduling feasibility, with limited attention to how flight- and service-time uncertainties propagate along reconnaissance–delivery task chains and lead to waiting and downstream delays. Meanwhile, existing two-stage stochastic models largely emphasize expected performance and provide limited treatment of tail risks such as severe delays and mission failures. To address these gaps, this study models flight and service times as scenario-dependent parameters, formulates waiting and delay as second-stage recourse costs, and incorporates Mean-CVaR into a risk-averse two-stage stochastic programming model.

From the solution perspective, existing Benders acceleration strategies typically target individual computational bottlenecks and are rarely designed for UAV task allocation problems that simultaneously involve integer routing, scenario-dependent time propagation, precedence relationships, and Mean-CVaR. To alleviate insufficient master problem information, slow initial solution search, and continuous cut growth, this study jointly introduces partial scenario embedding, a heuristic warm start, and dynamic cut-pool management to develop a problem-tailored Enhanced BD computational framework.

3. Two-Stage Stochastic Programming Model

3.1. Problem Description and Assumptions

This section develops a two-stage stochastic optimization framework for UAV task allocation in emergency rescue. Let N denote the set of disaster-affected points. Each disaster-affected point $n \in N$ corresponds to one reconnaissance task r_n and one delivery task e_n. A reconnaissance task is used to confirm the condition of the disaster-affected point, emergency-material demand, and local operating conditions. The corresponding delivery task transports first-aid kits, communication equipment, or other lightweight emergency supplies after the relevant information is confirmed. The two tasks form an information-acquisition and resource-response chain. To describe UAV routes, both reconnaissance and delivery tasks are treated as visitable nodes in the network. The reconnaissance and delivery task sets are denoted by $R = \{r_n \mid n \in N\}$ and $E = \{e_n \mid n \in N\}$, respectively, and the full task node set is $U = R \cup E$. The distribution center is denoted by node 0, and the emergency logistics network is therefore defined as $V = \{0\} \cup U$. If a UAV can fly directly from node i to node j, the ordered node pair (i, j) is included in the feasible node-pair set

A^F. In this network, multiple UAVs depart from the distribution center, visit their assigned task nodes, and return to the distribution center after completing their operations.

The model focuses on multi-type emergency tasks with temporal dependencies. Each delivery task can be performed only after the corresponding reconnaissance task has been completed. Accordingly, the service start time of each delivery task cannot be earlier than the completion time of its associated reconnaissance task. This precedence relation applies to the formal execution of delivery service at the disaster-affected point. Preparatory activities such as material loading, UAV standby, pre-positioning, and real-time information exchange may be conducted before the reconnaissance task is completed. UAV resource allocation jointly determines which UAV performs each task, the task composition of each UAV route, and the execution order of reconnaissance and delivery tasks.

The model further requires both the reconnaissance task node and the delivery task node associated with each disaster-affected point to be performed exactly once while satisfying routing feasibility constraints such as flow balance, endurance limits, and subtour elimination. Because flight and on-site service processes are uncertain in post-disaster environments, flight time and service time are treated as random parameters in a two-stage decision structure.

Before random parameters are revealed, the first stage determines task assignment, routes, and task sequences, which incur deterministic transportation costs. After a scenario is realized, the second stage keeps the first-stage route structure unchanged and describes the execution process of the fixed routing plan under different time-disturbance scenarios. It determines arrival time, service start time, and waiting time to represent scenario response, and then calculates the waiting and delay costs. Flight time and service time are represented by random parameters $t_{ij\omega}$ and $s_{i\omega}$, respectively. Waiting time is the interval between a UAV's arrival at a task node and the start of service. This modeling approach jointly describes ex ante routing decisions and ex post scenario responses, thereby reflecting the effect of time uncertainty on overall scheduling outcomes.

To highlight the model structure and maintain computational tractability, the following assumptions are made:

(1) The UAV fleet is homogeneous, and all UAVs participate in scheduling. Each task node is served only once and cannot be split or preempted. This assumption focuses the model on route allocation and temporal recourse; if heterogeneous fleets, task splitting, or task restructuring are considered, task matching, visit sequences, and resource allocation may change.

(2) Planning parameters such as task deadlines are deterministic; only flight time $t_{ij\omega}$ and service time $s_{i\omega}$ are scenario-dependent random variables represented by a discrete scenario set Ω. This assumption makes the results mainly reflect execution time fluctuations; if demand, resource availability, or deadlines are also uncertain, waiting, delay, and risk levels may change.

(3) Waiting cost is incurred only at task nodes U, and the distribution center node 0 does not incur waiting cost. The precedence relation "reconnaissance before delivery" must be strictly enforced. Waiting at task nodes is modeled as ground-idle waiting, which is applicable when UAVs can temporarily land or remain in a low-power ground-idle state at task nodes. Therefore, waiting time is not included in flight-endurance consumption. If aerial hovering is required in practice, this simplification may underestimate energy consumption and overestimate the feasibility of some routes.

(4) Apron capacity is assumed to be sufficient, and takeoff and landing congestion is ignored. This assumption does not explicitly model queueing delays caused by takeoff and landing resource competition; if apron capacity or takeoff/landing congestion is

considered, capacity, sequencing, and queueing constraints would be required, and departure times, route feasibility, and tail risk may be affected.

This study focuses on execution time disturbances and analyzes the effects of flight-time and service-time variations on waiting, delay, and tail risk in reconnaissance–delivery task chains.

Table 2 summarizes the notation used in this study.

Table 2. Notation.

Symbol	Description
Sets and indices	
N	Set of disaster-affected points
$V = \{0\} \cup U$	Set of nodes, where 0 denotes the distribution center and U denotes the set of task nodes
$R = \{r_n \mid n \in N\}$	Set of reconnaissance tasks
$E = \{e_n \mid n \in N\}$	Set of delivery tasks
$U = R \cup E$	Set of task nodes, including reconnaissance and delivery tasks
K	Set of UAVs; $k \in K$ denotes the UAV index, and $\lvert K \rvert$ denotes the number of UAVs
Ω	Discrete scenario set; $\omega \in \Omega$ denotes the scenario index, and $\lvert \Omega \rvert$ denotes the number of scenarios
A^F	Feasible arc set after preprocessing
Model parameters	
c_{unit}	Unit transportation cost coefficient (yuan/min)
t_{ij}	Baseline flight time between node i and node j (min)
s_i	Baseline service time of task node i (min)
C_{pen}	Unit delay penalty coefficient (yuan/min)
C_{wait}	Unit waiting cost coefficient (yuan/min)
L_i	Latest expected completion time of task node i (min)
T_{max}	Maximum endurance time of a UAV (min)
M_{ij}	Arc-specific big-M constant for arc (i, j) (min)
$t_{ij\omega}, s_{i\omega}$	Realized flight time and service time under scenario ω (min)
p_ω	Probability of scenario ω (dimensionless, in $[0, 1]$)
α, λ	Risk confidence level and risk preference coefficient (dimensionless, in $[0, 1]$)
Variables and functions	
x_{ijk}	Binary variable equal to 1 if UAV k travels from i to j, and 0 otherwise
$AT_{i\omega}$	Arrival time of node i under scenario ω
$ST_{i\omega}$	Service start time of node i under scenario ω
$WT_{i\omega}$	Waiting time at task node i under scenario ω
$D_{i\omega}$	Delay duration of task node i under scenario ω
θ_ω	Proxy variable for the second-stage recourse cost of scenario ω
η	Risk-threshold variable in CVaR linearization
z_ω	Excess-loss variable of scenario ω
$\mathcal{X}$	Feasible region of first-stage routing decisions
$\mathcal{F}(x, \omega)$	Constraint set for second-stage time variables under first-stage route x and scenario ω

3.2. Notation

For ease of model construction, Table 2 lists the main sets, model parameters, decision variables, and auxiliary notation.

3.3. Model Formulation

Based on the above system setting and notation, this section constructs the risk-neutral two-stage stochastic programming model (Risk-Neutral TSSP). The model minimizes the sum of first-stage deterministic transportation cost and second-stage expected recourse cost, and describes the execution process of emergency tasks through routing constraints, temporal propagation constraints, and precedence constraints.

3.3.1. Objective Function

The objective function consists of the first-stage transportation cost and the second-stage expected recourse cost:

$$\min \quad Z = C_1(x) + \mathbb{E}_\Omega[Q(x,\omega)] \tag{1}$$

In (1), x denotes the first-stage routing decision vector composed of all routing variables x_{ijk}, namely $x = (x_{ijk})_{(i,j)\in A^F, k\in K}$. Thus, the first-stage cost $C_1(x)$ is determined solely by the first-stage route structure. Because the route structure is determined before the random parameters are revealed, this cost remains unchanged across scenarios and mainly reflects the basic transportation consumption of UAV task execution. No additional fixed activation cost is introduced; the model focuses on the transportation cost induced by routing:

$$C_1(x) = \sum_{k\in K} \sum_{(i,j)\in A^F} c_{unit} \cdot t_{ij} \cdot x_{ijk} \tag{2}$$

where c_{unit} is the unit comprehensive transportation cost coefficient converted according to flight time. It converts UAV energy consumption, equipment depreciation, and operation and maintenance costs into a time-based transportation cost. t_{ij} denotes the baseline flight time between node i and node j.

The second-stage expected cost $\mathbb{E}_\Omega[Q]$ characterizes the average recourse cost under random scenarios, where $\mathbb{E}_\Omega[\cdot]$ is calculated according to scenario probabilities, i.e., $\sum_{\omega\in\Omega} p_\omega Q(x,\omega)$. When flight time and service time fluctuate, additional waiting and task delay may occur, forming the scenario-dependent recourse cost $Q(x,\omega)$. To avoid ambiguity caused by multiple UAVs returning to and departing from the distribution center, waiting cost is counted only at task nodes. The waiting and delay terms are introduced as scheduling-level penalties for late service. Although delays in emergency rescue may affect rescue effectiveness in a nonlinear manner, this study adopts a linear penalty function as a tractable approximation to preserve the linear structure of the second-stage subproblem, thereby facilitating model interpretation and subsequent solution. The scenario recourse cost is therefore defined as

$$Q(x,\omega) = \sum_{i\in U} \left(C_{pen} \cdot D_{i\omega} + C_{wait} \cdot WT_{i\omega}\right) \tag{3}$$

Given a first-stage routing plan x and scenario ω, the realized flight time $t_{ij\omega}$, service time $s_{i\omega}$, and temporal recursion constraints jointly determine the arrival time, service start time, waiting time $WT_{i\omega}$, and delay duration $D_{i\omega}$ of each task node. In this study, delay duration is measured by task completion time; a delay penalty is incurred when the task completion time exceeds the latest completion time.

3.3.2. Constraints

The model also includes routing and scheduling constraints to ensure feasibility:

$$\sum_{(i,j)\in A^F} t_{ij} \cdot x_{ijk} \le T_{max}, \quad \forall k \in K \tag{4}$$

$$\sum_{k\in K} \sum_{j\in V:(i,j)\in A^F} x_{ijk} = 1, \quad \forall i \in U \tag{5}$$

$$\sum_{j\in U:(0,j)\in A^F} x_{0jk} = 1, \quad \sum_{i\in U:(i,0)\in A^F} x_{i0k} = 1, \quad \forall k \in K \tag{6}$$

$$\sum_{i\in V:(i,h)\in A^F} x_{ihk} - \sum_{j\in V:(h,j)\in A^F} x_{hjk} = 0, \quad \forall h \in U, k \in K \tag{7}$$

$$ST_{i\omega} + s_{i\omega} + t_{ij\omega} - (1 - x_{ijk})M_{ij} \le AT_{j\omega}, \quad \forall (i,j) \in A^F,\ j \in U, \forall k \in K, \forall \omega \in \Omega \tag{8}$$

$$AT_{i\omega} \le ST_{i\omega}, \quad WT_{i\omega} = ST_{i\omega} - AT_{i\omega}, \quad \forall i \in U, \omega \in \Omega \tag{9}$$

$$ST_{r_n\omega} + s_{r_n\omega} \le ST_{e_n\omega}, \quad \forall n \in N, \forall \omega \in \Omega \tag{10}$$

$$D_{i\omega} \ge ST_{i\omega} + s_{i\omega} - L_i, \quad D_{i\omega} \ge 0, \quad \forall i \in U, \omega \in \Omega \tag{11}$$

$$\begin{aligned} &AT_{0\omega} = 0, \quad ST_{0\omega} = 0, \quad \forall \omega \in \Omega, \\ &AT_{i\omega} \ge 0, \quad ST_{i\omega} \ge 0, \quad WT_{i\omega} \ge 0, \quad \forall i \in U, \omega \in \Omega. \end{aligned} \tag{12}$$

$$\sum_{i\in S} \sum_{j\in S, j\ne i} x_{ijk} \le |S| - 1, \quad \forall S \subseteq U, |S| \ge 2, \forall k \in K \tag{13}$$

$$x_{ijk} \in \{0,1\}, \quad \forall (i,j) \in A^F, \forall k \in K \tag{14}$$

Constraints (4)–(7) describe first-stage routing feasibility. Constraint (4) limits the total flight time of each UAV over feasible node pairs A^F by its endurance limit and thereby provides a scheduling-level approximation of UAV endurance feasibility, without explicitly modeling task-payload variations or nonlinear battery discharge processes. Constraint (5) ensures that each task node is visited exactly once. Constraint (6) requires every UAV to depart from and return to the distribution center. Constraint (7) enforces flow balance at each task node. Constraints (8)–(12) handle temporal propagation and scenario uncertainty. Constraint (8) establishes the arrival time recursion on selected arcs. When UAV k travels from node i to task node j, the arrival time at node j cannot be earlier than the service start time at node i, the service time at node i, and the realized flight time from i to j. Constraint (9) defines waiting time over the task node set U. Under the above assumption, $WT_{i\omega}$ represents ground-idle waiting and affects only temporal propagation and waiting cost, rather than the endurance constraint in Constraint (4). Constraint (10) enforces the "reconnaissance before delivery" precedence relation. Constraint (11) defines the delay duration $D_{i\omega}$ as the nonnegative part of the task completion time $ST_{i\omega} + s_{i\omega}$ exceeding the latest completion time L_i. Constraint (12) sets the distribution center as the time origin and imposes nonnegativity restrictions on task node time variables. Since the distribution center does not involve an actual service operation, its service time is treated as zero ($s_{0\omega} = 0$) when the time recursion is applied from the distribution center to the first task node. Finally, Constraints (13) and (14) define integrality conditions. Constraint (13) is the subtour elimination constraint (SEC), which excludes closed task node cycles disconnected

from the depot. In the computational experiments, subtour elimination is implemented using MTZ ordering variables. Constraint (14) defines the binary routing variables.

4. Risk-Averse Two-Stage Stochastic Programming Model

This section presents the risk-averse two-stage stochastic programming model under the conditional value-at-risk (CVaR) criterion and reformulates it as a linear mixed-integer programming model. Section 4.3 then applies sample average approximation to discretize the model into a finite scenario set.

4.1. Risk-Averse Objective Function

To characterize tail risk under high-loss scenarios, conditional value-at-risk (CVaR) is introduced and combined with the expected recourse cost to construct a Mean-CVaR objective function. The expectation term describes average performance under ordinary scenarios, whereas the CVaR term measures high-loss risk in tail scenarios. By considering both components, the model balances average operational performance and tail risk control. The risk-averse two-stage stochastic programming model is expressed as

$$\min \quad C_1(x) + (1-\lambda)\mathbb{E}[Q(x,\omega)] + \lambda \mathrm{CVaR}_\alpha(Q(x,\omega)) \tag{15}$$

where $C_1(x)$ denotes the first-stage transportation cost and $Q(x,\omega)$ denotes the recourse cost generated under scenario ω after routing plan x is fixed. The recourse function $Q(x,\omega)$ represents the scenario-specific evaluation cost of a fixed routing plan x under scenario ω. It mainly captures the waiting and delay losses induced by stochastic flight time and service time, while the task assignment, route structure, and visiting order determined in the first stage remain unchanged. The term $\lambda \mathrm{CVaR}_\alpha(Q(x,\omega))$ is the tail risk penalty, which captures the influence of waiting and delay losses in high-recourse-cost scenarios at confidence level α.

The parameter $\lambda \in [0,1]$ is the risk-aversion coefficient, which adjusts the relative weights of the mean and CVaR terms in the objective function. When $\lambda = 0$, the model reduces to a risk-neutral expected-cost minimization model. As λ increases, tail high-loss scenarios receive stronger penalties, and the resulting solution becomes more conservative.

4.2. Linear Reformulation

Model (15) contains a CVaR term and is difficult to optimize directly. Auxiliary variables do not represent new scheduling decisions; they are used to express the tail loss in the CVaR term equivalently. Following the standard CVaR linearization method [7], a risk threshold variable η and an excess loss variable z_ω are introduced. Here, η denotes the risk threshold at confidence level α, and z_ω denotes the amount by which the recourse cost under scenario ω exceeds this threshold. Model (15) can then be reformulated as

$$\min \quad C_1(x) + (1-\lambda)\sum_{\omega\in\Omega} p_\omega Q(x,\omega) + \lambda\left(\eta + \frac{1}{1-\alpha}\sum_{\omega\in\Omega} p_\omega z_\omega\right) \tag{16}$$

To ensure that z_ω correctly represents tail excess loss, the following constraints are added:

$$z_\omega \geq Q(x,\omega) - \eta, \quad z_\omega \geq 0, \quad \forall \omega \in \Omega; \qquad \eta \geq 0 \tag{17}$$

When the recourse cost under scenario ω exceeds the threshold η, z_ω takes the excess amount; otherwise, $z_\omega = 0$. Thus, only excess losses in tail scenarios enter the CVaR calculation.

4.3. Sample Average Approximation Model

Because flight time and service time are continuous random variables, the scenario space of the risk-averse two-stage stochastic programming model is large and difficult to solve directly. The sample average approximation (SAA) method is used to discretize the original stochastic problem. Specifically, a discrete scenario set Ω is generated through Monte Carlo simulation. Each scenario corresponds to one realization of $t_{ij\omega}$ and $s_{i\omega}$. The expectation and risk terms in the original model are approximated by sample averages, yielding the following model:

$$\min_{x,\eta,z} \quad C_1(x) + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left[(1-\lambda) Q(x,\omega) + \frac{\lambda}{1-\alpha} z_\omega \right] + \lambda \eta \tag{18}$$

$$s.t. \quad x \in \mathcal{X}, \quad z_\omega \geq Q(x,\omega) - \eta, \quad z_\omega \geq 0, \quad \forall \omega \in \Omega; \qquad \eta \geq 0. \tag{19}$$

This formulation approximates the original stochastic problem using a finite scenario set. As the number of scenarios $|\Omega|$ increases, the approximation generally captures the uncertainty characteristics of the original problem more accurately.

5. Hybrid Benders Decomposition Algorithm

To address the difficulty of directly solving the risk-averse two-stage stochastic programming model under the tested medium-sized and largest configurations, a problem-tailored Enhanced BD computational framework is developed. This section presents the overall algorithmic idea, the master problem and subproblems, the acceleration strategies, and the complete algorithmic flow.

5.1. Algorithmic Idea

The first stage of the risk-averse two-stage stochastic programming model contains integer routing decisions and must handle many random scenarios. The proposed problem is NP-hard because it contains the deterministic vehicle routing problem (VRP) as a restricted special case. If stochastic scenarios, service time uncertainty, precedence-related recourse evaluation, and the Mean-CVaR term are removed, the problem reduces to a deterministic multi-UAV routing problem with the VRP structure. If the deterministic equivalent problem (DEP) is solved directly, scenario-dependent timing variables and constraints are replicated across all scenarios, causing the formulation to grow rapidly with the numbers of tasks and scenarios. A Benders-type decomposition method is therefore adopted to separate first-stage routing decisions from scenario-dependent recourse evaluation, combining scenario forward evaluation with dual information from continuous subproblems. Based on this decomposition structure, partial Benders decomposition, heuristic warm start, and dynamic cut-pool management are integrated to construct the Enhanced BD framework. The construction of the master problem and subproblems is presented first, followed by the three acceleration strategies and the overall algorithmic flow.

Algorithm 1 presents the pseudocode of the hybrid Benders decomposition algorithm, hereafter denoted as Enhanced BD. In the algorithm, x^ℓ denotes the candidate route obtained from the master problem at iteration ℓ, and x^* denotes the incumbent best feasible route retained during the iteration. The parameter q_p denotes the number of embedded scenarios in the partial Benders master problem, Ω_p is the embedded scenario set, and $\Omega_d = \Omega \setminus \Omega_p$ is the decomposed scenario set. The parameters δ, ϵ, and $I_{\max}$ denote the cut-violation threshold, convergence tolerance, and maximum number of iterations, respectively, while $P_{\max}$ and $K_{\max}$ denote the cut-pool capacity and the maximum number of cuts written back to the master problem in each iteration. LB and UB denote the global lower and upper bounds.

Algorithm 1 Hybrid Benders decomposition algorithm

Input: Scenario set Ω, risk parameters λ, α, embedded scenario number q_p, thresholds δ, ϵ, iteration limit $I_{\max}$, cut-pool parameters $P_{\max}, K_{\max}$.
Output: Routing solution x^* and objective value Z^*.
Construct embedded scenarios Ω_p using the service time burden rule; set $\Omega_d \leftarrow \Omega \setminus \Omega_p$; construct the partial Benders master problem MP.
Set $x^0 \leftarrow$ ConstructiveHeuristic().
Set $UB \leftarrow$ Evaluate(x^0, Ω) and $x^* \leftarrow x^0$.
Set $LB \leftarrow 0$ and $\ell \leftarrow 0$.
while $(UB - LB)/|UB| > \epsilon$ and $\ell < I_{\max}$ do
    $\ell \leftarrow \ell + 1$.
    Select historical strong cuts from the cut pool and add them to MP.
    Solve MP with warm start $x^{\ell-1}$, obtaining (x^ℓ, LB^ℓ); set $LB \leftarrow LB^\ell$.
    For each $\omega \in \Omega_d$, compute $Q_\omega \leftarrow$ ForwardSimulation(x^ℓ, ω).
    Set violated $\leftarrow \{\omega \in \Omega_d : Q_\omega > \theta_\omega^\ell + \delta\}$.
    if violated $\neq \emptyset$ then generate dual optimality cuts for violated scenarios, add selected cuts to MP, and store them in the cut pool.
    Set $UB^\ell \leftarrow$ Evaluate(x^ℓ, Ω).
    if $UB^\ell < UB$ then set $UB \leftarrow UB^\ell$ and $x^* \leftarrow x^\ell$.
end while
Return x^* and $Z^* \leftarrow UB$.

At each iteration, the master problem first generates a candidate routing solution. Forward simulation is then performed to evaluate the scenario-specific recourse costs of this candidate route and identify underestimated scenarios, for which Benders optimality cuts are generated. Finally, the candidate route is evaluated over the complete scenario set using the Mean-CVaR objective. If this evaluation yields a smaller upper bound, the algorithm updates UB and records the corresponding route as the new x^*. This process is repeated until convergence.

5.2. Master Problem and Subproblems

The original two-stage model contains first-stage discrete routing decisions and second-stage scenario-dependent timing decisions. As the number of tasks and scenarios increases, the model size expands rapidly and direct solution becomes increasingly difficult. A Benders decomposition framework is therefore adopted. The master problem determines first-stage routes, while the subproblems evaluate the second-stage recourse cost of each scenario under a given route. Information is exchanged through optimality cuts.

The master problem (MP) determines the first-stage route structure. Because the true recourse cost $Q(x, \omega)$ can be evaluated only after a route has been fixed and the scenario subproblem has been solved, the master problem uses θ_ω to represent the recourse cost under scenario ω and gradually approximates its true value through Benders cuts. To represent the Mean-CVaR risk term, the threshold variable η and auxiliary variables z_ω are introduced. At iteration ℓ, the master problem is formulated as

$$\min \quad Z_{MP}^\ell = C_1(x) + (1-\lambda) \sum_{\omega \in \Omega} p_\omega \theta_\omega + \lambda \left(\eta + \frac{1}{1-\alpha} \sum_{\omega \in \Omega} p_\omega z_\omega \right) \tag{20}$$

$$s.t. \quad x \in \mathcal{X} \tag{21}$$

$$z_\omega \geq \theta_\omega - \eta, \quad z_\omega \geq 0, \quad \theta_\omega \geq 0, \quad \forall \omega \in \Omega; \qquad \eta \geq 0 \tag{22}$$

Equations (20)–(22) embed the Mean-CVaR objective in the Benders master problem, and $\mathcal{X}$ in (21) follows the first-stage routing feasible region. Since the master problem

approximates $Q(x, \omega)$ only through the generated cuts, its objective value is usually no greater than the true optimal value of the complete model and can be used as the current lower bound LB. The master problem solution at iteration ℓ is denoted by x^ℓ.

When Partial Benders, warm start, and dynamic cut-pool management are disabled, the Basic BD benchmark used in this study is implemented as a scenario-wise multicut Benders baseline structurally aligned with the risk-averse multicut L-shaped method of Noyan [8]. Specifically, the master problem introduces a separate recourse approximation variable θ_ω for each scenario, and Benders optimality cuts are generated separately from the corresponding scenario subproblems. It is hereafter referred to as Basic BD. Enhanced BD builds on this baseline by further incorporating partial scenario embedding, heuristic warm start, and dynamic cut-pool management.

Given candidate route x^ℓ, the second-stage problem can be decomposed into independent scenario subproblems (SPs). Since the task visit sequence and UAV assignment are fixed, each subproblem optimizes only the time variables under scenario ω, including arrival time, service start time, waiting time, and delay, and then calculates the true recourse cost of that route:

$$Q(x^\ell, \omega) = \min \quad \sum_{i \in U} \left(C_{pen} D_{i\omega} + C_{wait} WT_{i\omega} \right) \tag{23}$$

$$s.t. \quad (AT_{i\omega}, ST_{i\omega}, WT_{i\omega}, D_{i\omega}) \in \mathcal{F}(x^\ell, \omega) \tag{24}$$

where $\mathcal{F}(x^\ell, \omega)$ denotes the set of constraints that the second-stage time variables must satisfy after first-stage route x^ℓ and scenario ω are given. This set includes time propagation, task precedence, and delay-definition constraints, corresponding to Constraints (8)–(12).

In Algorithm 1, ForwardSimulation(x^ℓ, ω) solves the scenario subproblem defined by Equations (23) and (24) for the fixed route x^ℓ under scenario ω and returns the corresponding recourse cost $Q(x^\ell, \omega)$. The operator Evaluate(x^ℓ, Ω) repeats this evaluation over the complete scenario set, substitutes the resulting recourse costs into the Mean-CVaR objective in Equations (16) and (17), and returns the feasible objective value used to update the global upper bound UB.

If a scenario satisfies

$$Q(x^\ell, \omega) > \theta_\omega^\ell + \delta, \tag{25}$$

the master problem underestimates the recourse cost of scenario ω, and the underestimation exceeds the tolerance threshold δ. A Benders optimality cut is then generated from the dual solution of the corresponding scenario subproblem:

$$\theta_\omega \geq \pi_{c,\omega}^\ell + \sum_{k \in K} \sum_{(i,j) \in A^F} \beta_{ijk\omega}^\ell \cdot x_{ijk}, \quad \forall \omega \in \text{violated} \tag{26}$$

where $\pi_{c,\omega}^\ell$ and $\beta_{ijk\omega}^\ell$ are computed from the dual solution of the scenario subproblem at iteration ℓ rather than specified as additional model parameters. The term $\pi_{c,\omega}^\ell$ is the constant term of the cut, and $\beta_{ijk\omega}^\ell$ is the coefficient of routing variable x_{ijk}, reflecting how different path choices affect the lower bound of recourse cost in that scenario. Equation (26) transforms the recourse-cost information from the current scenario into a linear constraint in x_{ijk} and returns it to the master problem to correct the underestimation of θ_ω.

5.3. Acceleration Strategies

Three acceleration strategies are further employed to improve solution efficiency for the tested medium-sized and largest configurations.

5.3.1. Partial Benders Decomposition

Even with dual cuts, many ineffective iterations may occur in the early stage of the algorithm when the emergency network topology is complex. To strengthen the master problem, the complete second-stage constraints of q_p embedded scenarios are incorporated into the master problem. In this study, Ω_p is constructed using a service time burden-driven rule rather than random sampling or Wasserstein distance-based scenario reduction. Specifically, all training scenarios are ranked in descending order according to the total realized service time $\sum_{i \in U} s_{i\omega}$. Scenarios with larger service time burdens are selected first. If the baseline scenario is not selected, it is retained in the embedded scenario set, and the remaining positions are filled by the highest-burden scenarios. The remaining scenarios constitute $\Omega_d = \Omega \setminus \Omega_p$. This rule is adopted because service time fluctuations directly affect task completion time, waiting time, and delay propagation along the reconnaissance–delivery task chains. Therefore, embedding high service burden scenarios enables the master problem to receive stronger timing and delay information in the early iterations. The set Ω_p is fixed after initialization and is not updated during the Benders iterations. In this way, the master problem performs time scheduling for these scenarios while making routing decisions:

$$\begin{aligned} \min \quad Z_{MP}^{partial} = C_1(x) + (1-\lambda)\left[\sum_{\omega \in \Omega_p} p_\omega Q(x,\omega) + \sum_{\omega \in \Omega_d} p_\omega \theta_\omega\right] \\ + \lambda\left(\eta + \frac{1}{1-\alpha}\sum_{\omega \in \Omega} p_\omega z_\omega\right). \end{aligned} \tag{27}$$

Correspondingly, the CVaR excess-loss variables must be constrained separately for embedded and decomposed scenarios:

$$\begin{aligned} z_\omega &\geq Q(x,\omega) - \eta, && \forall \omega \in \Omega_p, \\ z_\omega &\geq \theta_\omega - \eta, && \forall \omega \in \Omega_d, \\ z_\omega &\geq 0, && \forall \omega \in \Omega, \\ \eta &\geq 0. \end{aligned} \tag{28}$$

Here, Ω_p denotes the embedded scenario set, and $\Omega_d = \Omega \setminus \Omega_p$ denotes the decomposed scenario set. For $\omega \in \Omega_p$, the second-stage cost $Q(x,\omega)$ is calculated directly using timing variables in the master problem. For $\omega \in \Omega_d$, it is still represented by the proxy variable θ_ω. Thus, the master problem handles both route structure and partial timing information. To further evaluate the computational effect of this embedded-scenario selection rule, Section 6.2.3 compares it with random selection and Wasserstein distance-based representative selection under the same partial Benders setting.

5.3.2. Heuristic Warm Start

During Benders decomposition, the master problem must be solved repeatedly. Between two adjacent iterations, newly added information mainly consists of optimality or feasibility cuts, and the first-stage route structure usually changes only slightly. Solving each iteration from scratch may therefore cause substantial repeated computation. A heuristic warm-start strategy is adopted to improve master problem solution efficiency.

Specifically, an initial feasible route x^0 is constructed before the algorithm starts, and an initial upper bound UB is obtained through full-scenario evaluation. For general instances, the initial route is generated using a greedy heuristic. For the small-sized and medium-sized tested instances, a constructive routing heuristic is used. Reconnaissance tasks are first sorted according to their baseline travel time from the depot, and each reconnaissance task is

paired with its corresponding delivery task to preserve the reconnaissance-before-delivery precedence relation. These task pairs are then assigned to UAV routes in a round-robin manner. The generated route is checked against the first-stage routing constraints, including unique task visit, depot departure and return, flow balance, endurance, and reconnaissance–delivery pairing constraints. If the constructed route is infeasible, a lightweight first-stage routing MIP is solved to obtain a feasible route seed. For the largest tested instance, a deterministic VRP under the mean scenario is solved first, and the resulting route is used as the initial solution of the master problem. The mean scenario is constructed using the probability-weighted mean flight and service times, and the deterministic routing model retains the main routing, endurance, flow-balance, subtour-elimination, precedence, and timing constraints. In subsequent iterations, the optimal solution $x^{\ell-1}$ of the previous master problem is passed to the solver as a MIP start for the current master problem, accelerating the branch-and-bound search. The warm start only provides an incumbent routing solution to the solver and does not fix the routing variables.

Algorithm 2 summarizes this warm-start initialization procedure.

Algorithm 2 Heuristic warm-start initialization

Input: Task set U, UAV set K, and scenario set Ω.
Output: Initial route x^0 and initial upper bound UB.
if the instance is small-sized or medium-sized then
  $x^0 \leftarrow \text{ConstructiveRouteSeed}(U, K)$.
  if x^0 fails the first-stage feasibility check then $x^0 \leftarrow \text{SolveRouteSeedMIP}(U, K)$.
else $x^0 \leftarrow \text{SolveMeanScenarioVRP}(U, K, \Omega)$.
Set $UB \leftarrow \text{Evaluate}(x^0, \Omega)$.
Pass x^0 to the initial master problem as a MIP start.
In iteration ℓ, pass $x^{\ell-1}$ as the MIP start for the current master problem.
Return x^0 and UB.

5.3.3. Dynamic Cut-Pool Management with Scenario-Wise Multicut Generation

In small-scale instances, Benders optimality cuts usually tighten the master problem lower bound quickly and allow convergence within a small number of iterations. As the number of scenarios increases, scenario-wise cuts also increase, enlarging the master problem and reducing solution efficiency. To control this effect, dynamic cut-pool and multicut generation strategies are adopted to manage cut generation, storage, and reintroduction.

(1) Scenario-wise multicut generation. The CVaR objective is sensitive to high-loss tail scenarios. Therefore, the scenario-level proxy variable θ_ω is retained, and Benders optimality cuts are generated separately for underestimated scenarios. Compared with aggregating all scenario recourse costs into a single proxy variable, this treatment preserves cost differences across scenarios and reduces information loss caused by scenario aggregation.

(2) Cut-pool management and selection. Multicut generation increases the number of cutting planes in the master problem. To control model size, a dynamic cut pool with capacity $P_{max} = 150$ is used. In each iteration, only the top K_{max} historical or newly generated cuts with relatively large violations at the current solution x^ℓ are written into the master problem. The remaining cuts are stored in the cut pool, while cuts that have not been selected for a long time or contribute little are removed according to an aging rule.

(3) Progressive expansion of cut-generation scenarios. Each iteration still evaluates the current route over the full scenario set Ω to update the full-scenario upper bound UB and the relative gap $gap = (UB - LB)/|UB|$. To avoid generating many low-value cuts in early iterations, the algorithm first performs cut tests on a small active scenario subset. As the route structure stabilizes, the set of scenarios participating in dual verification is expanded. This treatment preserves full-scenario evaluation while reducing the computational burden of the master problem.

5.4. Algorithmic Flow

Based on the master problem, subproblems, and three acceleration strategies described above, Figure 1 presents the overall flow of the hybrid Benders decomposition algorithm. Partial Benders decomposition strengthens master problem information in early iterations, warm start shortens master problem solution time, and dynamic cut-pool management controls the number of cuts written back to the model.

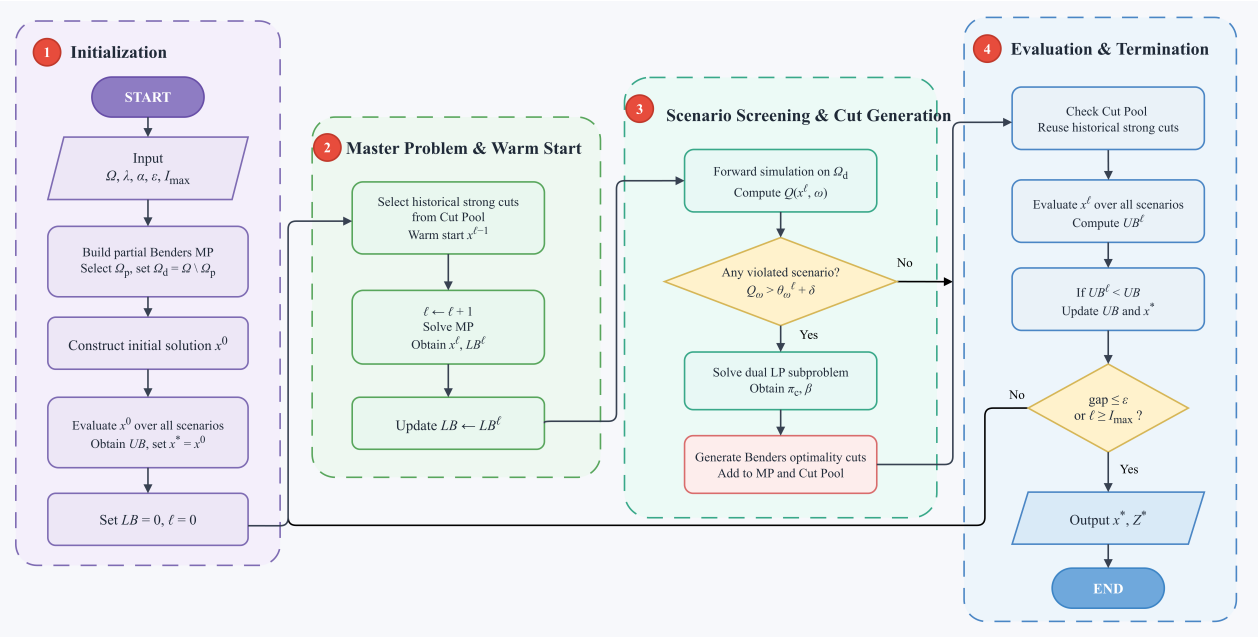


Figure 1. Flowchart of the hybrid Benders decomposition algorithm. The superscript * denotes the incumbent best solution.

6. Numerical Experiments

6.1. Experimental Settings

6.1.1. Instance Background

The experiments are based on an emergency rescue scenario in Jiangning District, Nanjing. The coordinates of the distribution center and disaster-affected points are taken from a previous UAV emergency distribution center location instance [49]. That instance identifies distribution centers and their service objects based on the geographic locations of candidate facilities and potential demand points, together with UAV coverage ranges. This study selects one distribution center and 16 disaster-affected points from that instance as the task allocation case. For each disaster-affected point $n \in N$, 1 reconnaissance task r_n and 1 delivery task e_n are defined, generating 32 task nodes in total. Each delivery task e_n can be executed only after its corresponding reconnaissance task r_n has been completed, which represents the operational requirement of reconnaissance before delivery. The computational experiments focus on coordinated multi-UAV operations. Different multi-UAV resource levels are examined through the fleet size sensitivity analysis in Section 6.4.2, while the Jiangning instance and the additional synthetic network topologies in Section 6.5 are used to evaluate the model under different spatial structures.

A UAV flight speed of 50 km/h is used to calculate baseline flight times. Deadlines for reconnaissance and delivery tasks are generated as baseline flight time plus service time plus a time margin. A 20 min margin is added for reconnaissance tasks; for delivery tasks, delivery service time and an additional 15 min margin are added based on the corresponding reconnaissance task. The coordinates and time parameters of the 16 disaster-affected points are reported in Table 3.

Table 3. Coordinates and time parameters of disaster-affected points.

Node	Longitude	Latitude	Distance (km)	Flight Time (min)	Recon. Deadline (min)	Delivery Deadline (min)
Distribution center						
0	118.857688	31.797570	0.0	0.0	–	–
1	118.8366	31.8487	6.0	7.2	30.2	50.2
2	118.8526	31.7754	2.5	3.0	26.0	46.0
3	118.9204	31.9052	13.4	16.0	39.0	59.0
4	118.9579	31.8033	9.5	11.4	34.4	54.4
5	118.7898	31.9134	14.4	17.3	40.3	60.3
6	118.8749	31.6934	11.7	14.0	37.0	57.0
7	118.7911	31.7180	10.9	13.0	36.0	56.0
8	118.8397	31.9094	12.5	15.1	38.1	58.1
9	118.7905	31.7829	6.6	7.9	30.9	50.9
10	118.7950	31.8377	7.4	8.9	31.9	51.9
11	118.9385	31.8388	8.9	10.7	33.7	53.7
12	118.8779	31.8053	2.1	2.5	25.5	45.5
13	118.8031	31.8811	10.6	12.7	35.7	55.7
14	118.8877	31.7711	4.1	4.9	27.9	47.9
15	118.8944	31.8835	10.2	12.2	35.2	55.2
16	118.9380	31.7007	13.2	15.8	38.8	58.8

6.1.2. Parameter Settings and Data Sources

This section reports the experimental parameters and their basis. The parameters include basic operational and solution parameters, as well as uncertainty parameters, as shown in Tables 4 and 5. Flight speed and endurance time are mainly based on existing UAV application studies. Service time, fleet size, number of scenarios, risk parameters, and time-window margins are set according to the task process, and their effects are further examined in the sensitivity analysis.

Table 4 lists the basic operational and solution parameters. The UAV flight speed is set to 50 km/h, from which baseline flight times are calculated. The deadlines of reconnaissance and delivery tasks are determined according to "baseline flight time + service time + time margin." A 20 min time margin is added to reconnaissance tasks, and delivery tasks further include delivery service time and a 15 min time margin based on the corresponding reconnaissance task. Service times are measured in minutes and reflect the operational differences between reconnaissance hovering and material delivery.

Table 4. Basic operational and solution parameter settings.

Parameter	Value
UAV flight speed	50 km/h
Maximum endurance time T_{max}	40 min
Mean reconnaissance service time μ_s^R	3 min
Mean delivery service time μ_s^D	5 min
Unit transportation cost c_{unit}	2 yuan/min
Delay penalty coefficient C_{pen}	10 yuan/min
Waiting cost coefficient C_{wait}	1 yuan/min
Number of UAVs $\|K\|$	8
Number of training scenarios $\|\Omega_{\text{train}}\|$	30
Confidence level α	0.95
Risk preference coefficient λ	0.5
Convergence tolerance ϵ	5%

Table 5. Uncertainty parameter settings.

Parameter	Distribution	Parameter Values	Basis
Flight time $t_{ij\omega}$	Truncated normal $TN(t_{ij}, (0.15\,t_{ij})^2)$	Lower bound $0.7\,t_{ij}$; upper bound $1.5\,t_{ij}$	The parameterization uses a 15% coefficient of variation and a right-skewed truncation interval to represent bounded flight-time disturbances.
Service time $s_{i\omega}$	Truncated normal $TN(s_i, (0.2\,s_i)^2)$	Lower bound $0.6\,s_i$; upper bound $1.8\,s_i$	A 20% coefficient of variation is used because on-site operational variability is assumed to exceed flight-time variability.

The computational environment and stopping criterion were set as follows. All computational experiments were conducted on a MacBook Pro equipped with an Apple M3 processor with 8 CPU cores and 8 GB of memory, running macOS 26.4.1. The algorithms were implemented in Python 3.12.6 and solved using Gurobi 13.0.1 with four solver threads.

A wall-clock limit of 600 s was applied to DEP, Basic BD, and Enhanced BD as a common computational budget. The same limit was applied to each method on the same instances and scenario sets under identical computational settings. The small-sized instances terminated within the limit, while the medium-sized and largest tested instances were evaluated using the best available solution and convergence information obtained within 600 s. For non-convergent runs, termination may also occur before the wall-clock limit when the prescribed iteration limit is reached; therefore, runtime should be interpreted together with the final BD Gap and the number of successful runs.

To account for the randomness introduced by Monte Carlo scenario generation, each core computational configuration was independently repeated 10 times using random seeds, 101–110. In each replication, a new training scenario set was generated, and DEP, Basic BD, and Enhanced BD were evaluated on the same scenario set for that seed. This paired design ensures that performance differences among methods are not caused by different scenario samples. The objective value, runtime, iteration count, and final BD Gap are reported as mean ± standard deviation over the ten replications. For runs that did not meet the convergence criterion within the 600 s time limit, the incumbent objective value and final gap at termination were recorded when available, and the number of successful runs is reported separately.

The effect of the training sample size is further examined through repeated SAA experiments and independent out-of-sample evaluation in Section 6.4.1.

Table 5 reports the uncertainty parameter settings. Only stochastic fluctuations in flight time and service time are considered. The flight time $t_{ij\omega}$ reflects transport-time variations caused by weather changes, temporary detours, and short-term airspace restrictions, whereas the service time $s_{i\omega}$ reflects operational-time variations caused by information confirmation, on-site reconnaissance, material delivery, and related handling activities. Because both time components fluctuate around baseline values and unrealistic extreme samples should be avoided, truncated normal distributions are used. Their means are set to the baseline flight time and baseline service time, respectively.

Specifically, flight time is set as $t_{ij\omega} \sim TN(t_{ij}, (0.15\,t_{ij})^2)$, and service time is set as $s_{i\omega} \sim TN(s_i, (0.2\,s_i)^2)$. A 15% coefficient of variation for flight time represents a moderate disturbance level, while a 20% coefficient of variation for service time reflects stronger uncertainty in on-site operations than in flight. To avoid extreme samples outside realistic ranges, flight time is truncated to $[0.7\,t_{ij}, 1.5\,t_{ij}]$, and service time is truncated to $[0.6\,s_i, 1.8\,s_i]$.

For scenario generation, Monte Carlo sampling is used to construct the discrete scenario set Ω, and the model is solved under the SAA framework. The number of scenarios reflects a trade-off between the accuracy of uncertainty characterization and computational burden. Small-scale ablation experiments use 10 scenarios, the benchmark training stage uses 30 scenarios, and solvability tests are extended to 50 and 100 scenarios. In the ex-

tended cross-scale ablation experiment, the medium-sized and largest tested configurations use the same 50- and 100-scenario settings as the corresponding tested size performance experiments. The SAA training sample size analysis evaluates fixed routing solutions using 1000 independent out-of-sample scenarios to reduce the dependence of tail risk evaluation on only a few extreme scenarios, whereas the risk preference analysis uses 100 independent out-of-sample scenarios. The effect of common flight time disturbances across arcs is further examined through the correlated scenario sensitivity analysis in Section 6.4.6.

6.2. Solution Performance

This section evaluates the computational performance of Enhanced BD. First, the solution performance of DEP, Basic BD, and Enhanced BD is compared across different tested instance sizes. An ablation experiment is then conducted to examine the relative roles of the acceleration strategies in the convergence process.

6.2.1. Computational Performance Across Tested Instance Sizes

To examine solution performance as the tested configuration increases, small-sized, medium-sized, and largest tested instances are constructed, and Enhanced BD is compared with the deterministic equivalent problem (DEP) and Basic BD, the scenario-wise multicut Benders baseline structurally aligned with Noyan's risk-averse multicut L-shaped method [8]. The small-sized instance is used mainly to verify consistency between Enhanced BD and the deterministic equivalent model. The medium-sized instance compares convergence stability across methods, and the largest tested instance tests solvability when direct solution becomes difficult.

Table 6 summarizes the repeated computational results under different tested sizes. The small-sized, medium-sized, and largest tested instances correspond to different configurations of demand points, UAVs, and scenarios. Values are reported as mean ± standard deviation over ten independent replications. "Successful runs" denotes the number of replications that reached the prescribed convergence criterion within the 600 s time limit. Because DEP directly solves the deterministic equivalent model, BD Gap and Benders iteration count are not applicable to DEP. For non-convergent runs, the reported objective value and final gap correspond to the incumbent solution and termination state when available.

Table 6 shows that repeated scenario generation affects the objective value, runtime, and final BD Gap, but does not change the overall performance ranking among the tested methods. Enhanced BD converges in all ten replications for the small-sized and medium-sized instances and reaches the convergence criterion in eight out of ten replications for the largest tested instance, showing stronger stability across independently generated scenario sets. In contrast, Basic BD achieves only 1/10, 0/10, and 0/10 successful runs across the three scales, respectively. DEP converges in only two medium-sized replications and does not return stable valid objective values in the largest tested instance.

In the small-sized instance, DEP and Enhanced BD both converge in all ten replications and obtain close average objective values, indicating that the decomposition framework preserves good solution quality. In the medium-sized instance, Basic BD has a shorter average runtime, but its average final BD Gap reaches 132.16%, and none of the ten replications satisfy the convergence criterion. Thus, the shorter runtime of a non-convergent method should not be regarded as a computational advantage but together with the final BD Gap and the number of successful runs. In the largest tested instance, neither DEP nor Basic BD converge in any of the ten replications, whereas Enhanced BD converges in eight replications and obtains an average final BD Gap of 10.80%. Although the final gap still varies across random scenario sets, Enhanced BD remains more solvable in most replications.

Table 6. Computational performance across tested instance sizes under repeated scenario generation.

Scale	Method	Objective Value	Time (s)	Iterations	BD Gap (%)	Successful Runs
Small-sized (6/3/10)	DEP	120.93 ± 2.16	0.59 ± 0.12	N/A	N/A	10/10
Small-sized (6/3/10)	Basic BD	125.47 ± 5.79	3.04 ± 0.42	19.60 ± 1.26	246.92 ± 227.24	1/10
Small-sized (6/3/10)	Enhanced BD	120.38 ± 1.78	0.61 ± 0.50	2.30 ± 1.25	0.00 ± 0.00	10/10
Medium-sized (10/5/50)	DEP	248.10 ± 19.78	564.45 ± 76.24	N/A	N/A	2/10
Medium-sized (10/5/50)	Basic BD	609.08 ± 8.94	19.39 ± 2.28	10.00 ± 0.00	132.16 ± 18.77	0/10
Medium-sized (10/5/50)	Enhanced BD	225.55 ± 1.75	57.75 ± 22.10	1.50 ± 0.85	0.00 ± 0.00	10/10
Largest tested (16/8/100)	DEP	N/A	601.66 ± 0.49	N/A	N/A	0/10
Largest tested (16/8/100)	Basic BD	602.27 ± 16.63	392.47 ± 0.11	6.00 ± 0.00	152.90 ± 72.23	0/10
Largest tested (16/8/100)	Enhanced BD	449.21 ± 5.89	129.31 ± 1.65	8.70 ± 0.95	10.80 ± 8.55	8/10

Note: Values are reported as mean ± standard deviation over ten independent replications. Iterations are averaged over replications and therefore may be non-integer. Successful runs indicate the number of replications that reached the prescribed convergence criterion within the 600 s time limit. DEP directly solves the deterministic equivalent problem; therefore, BD Gap and Benders iteration count are not applicable. For non-convergent DEP runs, the objective value is reported only when a feasible incumbent is available. Since DEP fails to return valid objective values in most largest-tested replications, its objective value is reported as N/A. For non-convergent BD-type runs, a shorter runtime should not be interpreted as a computational advantage by itself, but should be considered together with the final BD Gap and the number of successful runs.

Figure 2 presents representative BD Gap trajectories for the medium-sized and largest tested instances, illustrating the early-stage bound-gap reduction process. The figure is used to show typical convergence behavior, while the statistical comparison across ten independent replications is reported in Table 6. In these representative runs, Enhanced BD reduces the BD Gap earlier than Basic BD, which is consistent with the higher convergence success rate and lower final BD Gap reported in Table 6.

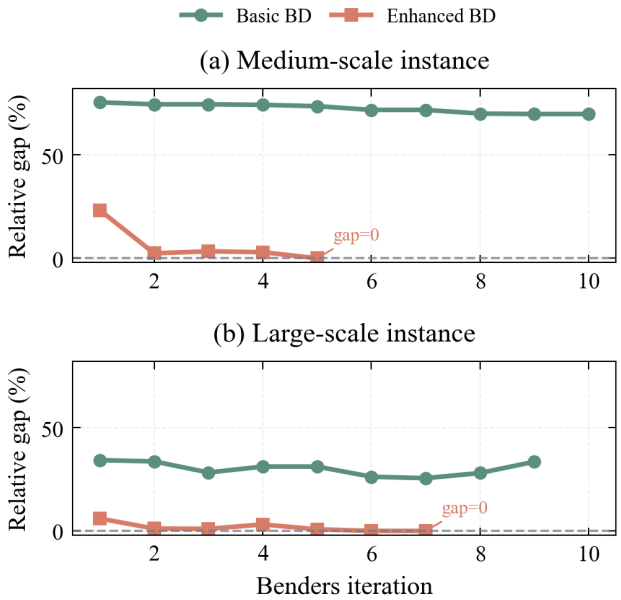


Figure 2. Early-stage BD Gap changes under different algorithms.

Overall, the repeated experiments show that the main advantage of Enhanced BD lies in its higher convergence success rate, lower terminal gap, and better convergence performance over the tested configurations, rather than in achieving the shortest runtime in every configuration. The results also confirm that directly solving DEP becomes unreliable as the tested configuration grows.

6.2.2. Ablation Experiment on Acceleration Strategies

To examine whether the relative roles of the three acceleration strategies vary with instance size, the ablation experiment is extended from the original small-sized instance to the medium-sized and largest tested instances. Five configurations are compared under the same instance data, scenario samples, risk parameters, and computational settings: Basic BD, Partial Benders only, Warm Start only, Dynamic Cut Pool only, and Enhanced BD with all three strategies combined.

Table 7 shows that partial Benders remains the most influential single acceleration component across the tested scales. In the medium-sized instance, partial Benders alone closes the BD gap within 221.4 s, whereas Warm Start only and Dynamic Cut Pool only remain non-convergent under the same computational budget. In the largest tested instance, partial Benders alone improves the bound earlier than the other single-strategy variants, but it does not satisfy the full convergence check within the common time limit. Warm Start only and Dynamic Cut Pool only still retain substantially larger gaps.

Table 7. Cross-scale ablation results of algorithmic acceleration strategies.

Scale	Basic BD	Partial Benders Only	Warm Start Only	Dynamic Cut Pool Only	Enhanced BD
Small-sized (6/3/10)	4.9 s 83.02%	1.5 s 0.00%	5.4 s 299.01%	4.8 s 53.67%	0.3 s 0.00%
Medium-sized (10/5/50)	589.3 s 35.12%	221.4 s 0.00%	588.0 s 35.19%	596.9 s 34.91%	333.3 s 0.00%
Largest tested (16/8/100)	575.7 s 33.47%	579.5 s 0.00% *	575.9 s 34.92%	577.2 s 34.92%	131.7 s 0.00%

Note: Values are reported as solution time (s)/final BD gap. For non-convergent BD-type runs, the reported value corresponds to the last completed Benders iteration within the 600 s wall-clock limit. The asterisk indicates that although the displayed bound gap reaches 0.00% at the last retained iteration, the full convergence check is not satisfied within the time limit.

Figure 3 visualizes the same cross-scale ablation results reported in Table 7.

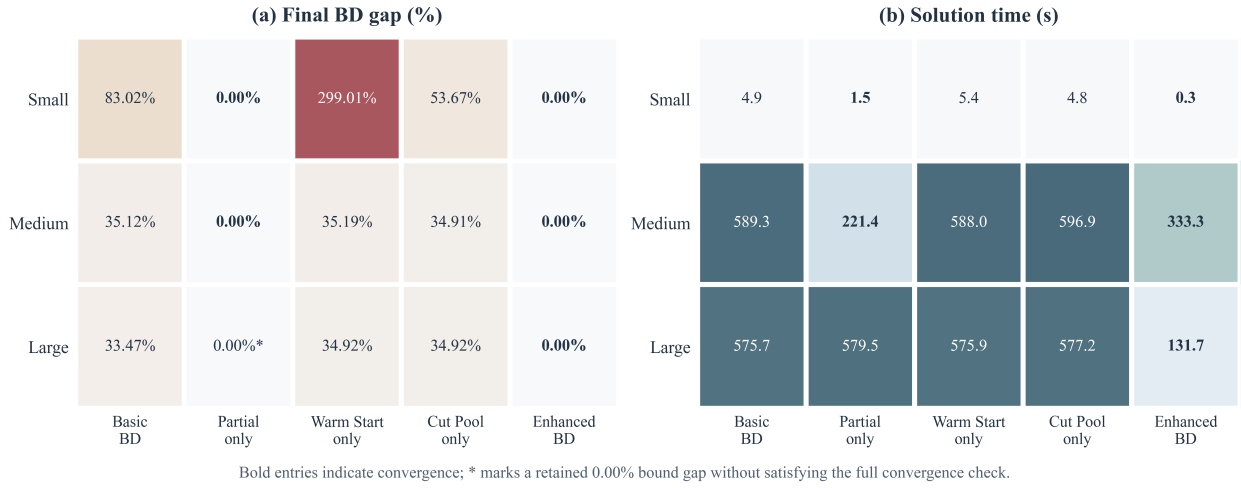


Figure 3. Cross-scale ablation results of Benders acceleration strategies. In panel (**a**), darker red indicates a larger final BD gap; in panel (**b**), darker blue indicates a longer solution time. Bold entries indicate convergence.

These results also show why the complete Enhanced BD framework remains necessary. Partial Benders mainly improves the quality of the master problem approximation by embedding high service burden scenarios, but in the largest tested instance it is not sufficient by itself to achieve full convergence. When partial Benders is combined with warm start and dynamic cut-pool management, Enhanced BD closes the BD gap and reaches a convergent solution in the largest tested instance. Thus, Warm Start and Dynamic Cut Pool are not dominant when used alone, but they provide supporting mechanisms in the combined framework.

6.2.3. Comparison of Embedded-Scenario Selection Rules

To directly evaluate the embedded-scenario selection rule used in partial Benders, three selection rules are compared on the medium-sized instance with three independent scenario replications: random selection, Wasserstein distance-based representative selection, and the proposed service time burden rule. All other algorithmic components, including warm start, dynamic cut-pool management, the number of embedded scenarios, risk parameters, solver settings, and computational budget, are kept unchanged. The purpose of this experiment is not to claim that the service time burden rule is generally superior to Wasserstein-based scenario selection but to examine whether it better strengthens the partial Benders master problem for the present task chain scheduling problem.

Table 8 shows that all three rules reach the convergence criterion in the medium-sized instance, but the service time burden rule requires fewer Benders iterations, fewer generated cuts, and less runtime. Random selection provides a structure-free baseline, while Wasserstein distance-based selection emphasizes distributional representativeness. In contrast, the proposed rule embeds scenarios with larger total service time burdens, which are more likely to create tight timing constraints, waiting, and delay propagation in reconnaissance–delivery task chains. Therefore, in the tested instance, this problem-specific rule strengthens the partial Benders master problem more efficiently than the two alternative selection criteria.

Table 8. Comparison of embedded scenario selection rules in partial Benders.

Selection Rule	Runtime (s)	Iterations	Generated Cuts	BD Gap (%)	Successful Runs
Random selection	119.07 ± 46.80	3.67 ± 1.53	73.33 ± 47.01	0.00 ± 0.00	3/3
Wasserstein distance-based selection	140.27 ± 44.43	4.33 ± 1.53	81.67 ± 23.18	0.00 ± 0.00	3/3
Service time burden rule	51.33 ± 15.83	1.33 ± 0.58	8.33 ± 14.43	0.00 ± 0.00	3/3

Note: Values are reported as mean ± standard deviation over three independent scenario replications of the medium-sized instance. The number of embedded scenarios is fixed at five for all rules. The final BD gap and successful runs are reported to verify that the comparison reflects convergence behavior rather than early termination.

6.2.4. Value of the Stochastic Solution and Expected Value of Perfect Information Analysis

To quantify the benefit of explicitly modeling time uncertainty relative to a deterministic average-scenario solution, this section evaluates the Value of the Stochastic Solution (VSS). In response to the reviewer's comment, the Expected Value of Perfect Information (EVPI) is also calculated under the same risk-neutral setting with $\lambda = 0$. The analysis is conducted under the risk-neutral setting with $\lambda = 0$ to isolate the value of stochastic modeling from the additional effect of Mean-CVaR risk aversion. This setting is used only for the VSS and EVPI analysis, while the baseline risk-averse experiments retain $\lambda = 0.5$.

The expected-value problem is constructed by replacing the stochastic flight and service times with their probability-weighted means. Solving this deterministic problem yields the first-stage routing solution x^{EV}. The route x^{EV} is then fixed and evaluated over the same stochastic scenario set and scenario probabilities used in the risk-neutral stochastic

model, producing the expected result of the expected-value solution, denoted by EEV. Let RP denote the optimal objective value of the risk-neutral stochastic programming model over the same scenario set. For EVPI, a wait-and-see benchmark is further constructed. In each scenario ω, the flight and service times of that scenario are assumed to be known before the routing decision is made, and the corresponding deterministic problem is solved to obtain Z_{ω}^{WS}.

$$\bar{t}_{ij} = \sum_{\omega \in \Omega} p_{\omega} t_{ij\omega}, \qquad \bar{s}_i = \sum_{\omega \in \Omega} p_{\omega} s_{i\omega}.$$

$$EEV = C_1(x^{EV}) + \sum_{\omega \in \Omega} p_{\omega} Q(x^{EV}, \omega).$$

$$VSS = EEV - RP, \qquad VSS_{\text{rel}} = \frac{EEV - RP}{RP} \times 100\%.$$

$$WS = \sum_{\omega \in \Omega} p_{\omega} Z_{\omega}^{WS}.$$

$$EVPI = RP - WS, \qquad EVPI_{\text{rel}} = \frac{RP - WS}{RP} \times 100\%.$$

Table 9 reports the VSS and EVPI results for the small-scale benchmark instance. The risk-neutral stochastic model, deterministic expected-value problem, and wait-and-see deterministic problems are solved under the same scenario set and $\lambda = 0$ setting.

Table 9. Value of the stochastic solution and expected value of perfect information.

Indicator	Description	Value
WS	Expected objective value under perfect information	111.81
RP	Optimal objective value of the risk-neutral stochastic model	118.73
EEV	Expected performance of the expected-value solution	119.31
$EVPI$	$RP - WS$	6.93
Relative EVPI	$EVPI/RP \times 100\%$	5.83%
VSS	$EEV - RP$	0.58
Relative VSS	$VSS/RP \times 100\%$	0.49%

Table 9 shows that $WS < RP < EEV$, which is consistent with the theoretical ordering for a minimization problem. The positive EVPI indicates that perfect information on scenario-dependent flight and service times can further reduce the objective value. However, EVPI represents an ideal benchmark rather than an implementable scheduling policy, because such complete scenario information is unavailable before dispatch in practical emergency operations. The positive VSS indicates that the routing solution obtained from the deterministic average-scenario problem incurs a higher expected cost when evaluated over the original stochastic scenarios. Explicitly modeling flight time and service time uncertainty reduces the expected cost by 0.58, corresponding to a relative improvement of 0.49%. Although the relative VSS is small, its positive value shows that the stochastic model provides a measurable improvement over relying only on mean parameter values. Together, EVPI and VSS show that execution time uncertainty has measurable decision value while explicitly modeling stochastic scenarios improves solution quality relative to relying only on mean parameter values.

6.3. Analysis of the Optimized Scheduling Solution

This section analyzes the optimized scheduling solution obtained by Enhanced BD under the benchmark settings in Tables 3–5, including 16 disaster-affected points, 8 UAVs,

30 training scenarios, $\alpha = 0.95$, and $\lambda = 0.5$. The analysis focuses on spatial routes, task chain structure, objective value decomposition, and sources of delay risk.

6.3.1. Task Allocation Scheme

Figure 4 shows the optimized scheduling solution for the benchmark instance. Figure 4a presents the spatial flight paths of the UAVs, and Figure 4b shows the corresponding reconnaissance–delivery task chains. Hollow boxes denote reconnaissance tasks, filled boxes denote delivery tasks, and dashed links indicate the precedence pairs for the same disaster-affected point. The figure shows that all eight UAVs are used, but their workloads are uneven. Some reconnaissance and delivery tasks are performed consecutively, whereas other delivery tasks are executed after intervening tasks. This indicates that the optimized schedule does not simply bind each reconnaissance–delivery pair independently; instead, it forms chained UAV schedules under the combined effects of routing cost, endurance constraints, and precedence relationships.

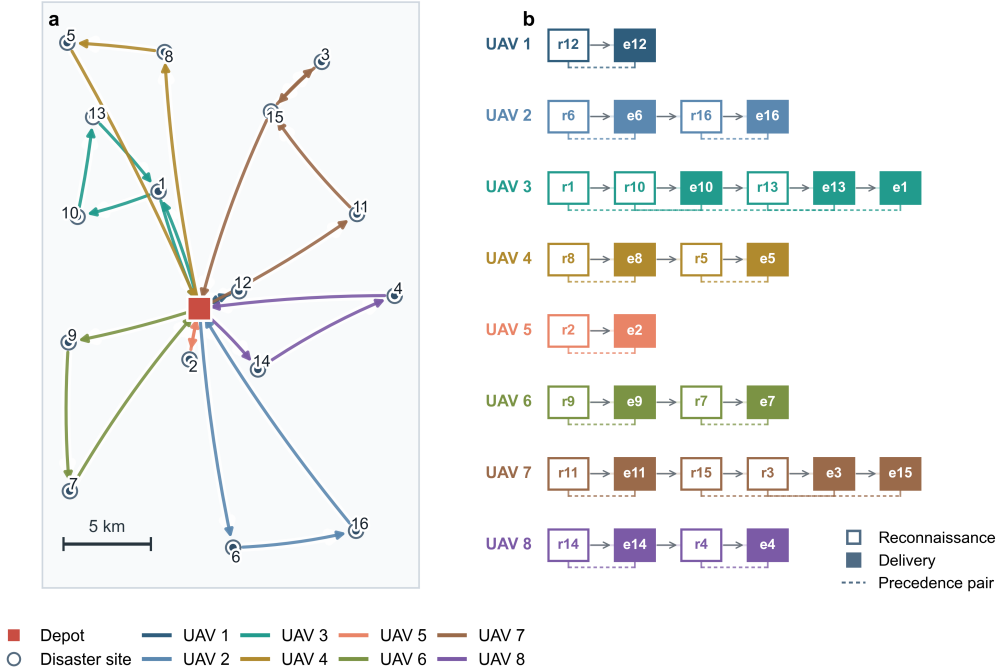


Figure 4. Optimized UAV scheduling solution: (**a**) spatial flight paths; (**b**) reconnaissance–delivery task chains.

6.3.2. Analysis of Risk Sources

Based on the scheduling solution shown in Figure 4, this study calculates the objective-value decomposition and scenario-dependent recourse costs of the solution. In Table 10, the first-stage transportation cost is obtained from the flight-time cost of the fixed schedule. The expected recourse cost and $\mathrm{CVaR}_{0.95}$ are calculated from waiting and delay costs under 30 training scenarios.

Table 10. Cost decomposition and objective value of the optimized scheduling solution.

Cost Component	Value (yuan)
First-stage transportation cost C_1	421.93
Expected recourse cost $\mathbb{E}[Q]$	1.11
$\mathrm{CVaR}_{0.95}[Q]$	13.78
$(1-\lambda)\mathbb{E}[Q]$	0.56
$\lambda \cdot \mathrm{CVaR}_{0.95}[Q]$	6.89
Total objective value Z^*	429.38

Table 10 shows that $\mathrm{CVaR}_{0.95}[Q]$ is substantially higher than the expected recourse cost, indicating that recourse costs are concentrated in a small number of high-loss scenarios. A comparison of the weighted terms further shows that $\lambda \mathrm{CVaR}_{0.95}[Q]$ is higher than $(1-\lambda)\mathbb{E}[Q]$. Thus, the second-stage risk cost of this solution is mainly driven by tail scenarios, while the share of average additional cost is relatively low. In emergency UAV scheduling, evaluating a solution only by expected recourse cost may therefore underestimate the impact of extreme delays on task reliability. Figure 5 presents the demand-point-scenario delay risk distribution to identify the demand points and scenarios associated with tail losses.

Figure 5a shows the distribution of the 16 demand points across delay scenarios. The horizontal axis gives the training scenario numbers with nonzero delay contributions, arranged in ascending order, and the vertical axis gives the demand point number. Light-colored cells indicate no delay contribution in the corresponding scenario, whereas orange-red cells indicate positive delay contributions. Darker colors represent higher delay levels. Delay risk is mainly concentrated at Sites 1, 3, and 15, while the other demand points show limited delay contributions in these scenarios.

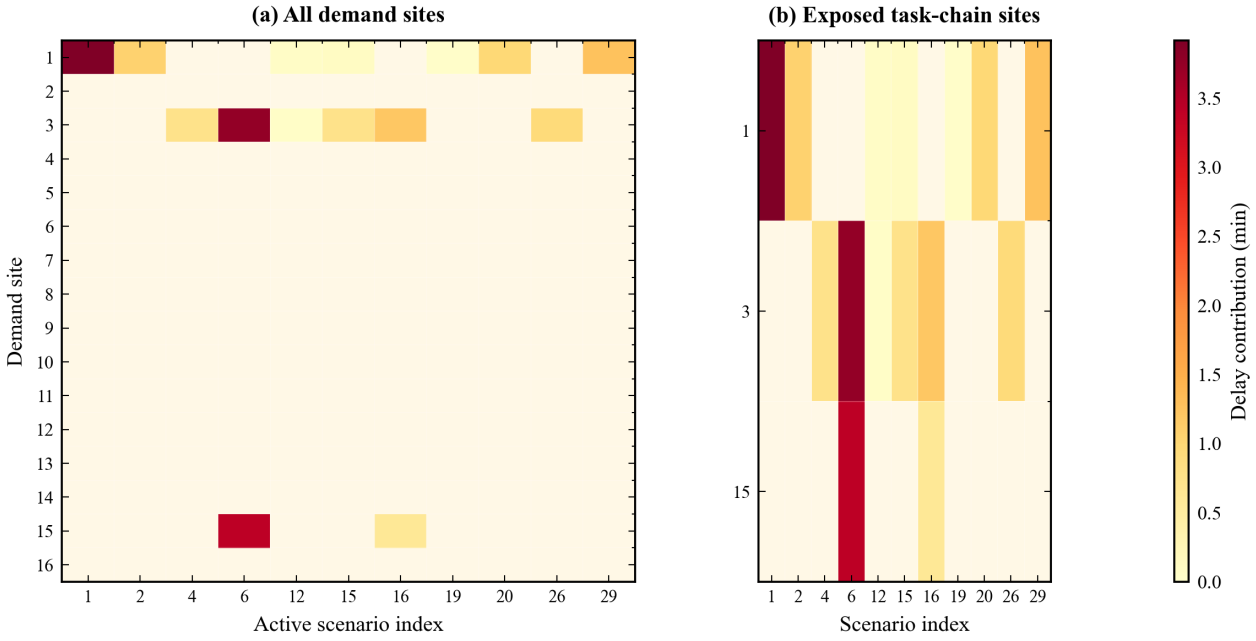


Figure 5. Demand point scenario delay risk distribution.

Figure 5b shows that tail risk is mainly concentrated at downstream nodes in two task chains. Site 1 belongs to the task chain of UAV 3, $r1 \rightarrow r10 \rightarrow e10 \rightarrow r13 \rightarrow e13 \rightarrow e1$, where $e1$ is the terminal task of the chain. Sites 3 and 15 belong to the task chain of UAV 7, $r11 \rightarrow e11 \rightarrow r15 \rightarrow r3 \rightarrow e3 \rightarrow e15$, where $e3$ and $e15$ also appear relatively late in the visit order. This pattern indicates that although longer task chains can improve route utilization, downstream tasks accumulate uncertainty from preceding flight and service processes. When preceding tasks take longer in some scenarios, the start times of downstream tasks are postponed, making deadline violations more likely.

Combining the delay distribution in Figure 5 with the task chains in Figure 4, the delay risk of the current solution is mainly concentrated at a small number of sites and the downstream nodes of their task chains. Sites 1, 3, and 15 have relatively high delay contributions, and downstream tasks are more vulnerable to the accumulated time of preceding reconnaissance and delivery tasks.

6.4. Parameter Sensitivity Analysis

The preceding sections analyze the model and algorithm from the perspectives of solution performance and scheduling solutions. This section further examines the effects of training sample size, fleet size, confidence level, risk preference coefficient, fluctuations in flight and service times, and correlated flight-time disturbances. The SAA sample size analysis evaluates the stability of stochastic estimates as the number of training scenarios increases. Fleet size and confidence level are analyzed mainly using model results under training scenarios, whereas the risk preference coefficient, time fluctuations, and correlated flight-time disturbances are compared using out-of-sample performance under unseen scenarios.

6.4.1. Sensitivity to the SAA Training Sample Size and Out-of-Sample Performance

To examine how the number of training scenarios affects the SAA solution and its out-of-sample performance, four settings, $N \in \{30, 50, 100, 200\}$, are considered on the same benchmark instance with $\lambda = 0.5$ and $\alpha = 0.95$. For each setting, five independent training samples are generated using different random seeds and solved by Enhanced BD under the same 600 s computational budget. Each resulting first-stage routing solution is then evaluated using a common set of 1000 independently generated out-of-sample scenarios.

At $\alpha = 0.95$, the nominal number of observations associated with the upper 5% tail increases from 1.5 for $N = 30$ to 10 for $N = 200$. A larger training sample therefore provides more observations for empirical tail risk estimation, but it also enlarges the multi-scenario optimization problem. The results are consequently interpreted by jointly considering the statistical indicators and the final optimization gaps. Table 11 reports the mean training objective, the standard deviation of the training objective, the mean out-of-sample variance, the mean out-of-sample $\mathrm{CVaR}_{0.95}$, and the mean optimality gap across the five replications.

Table 11. Effect of the SAA training sample size.

Training Scenarios	Mean Training Objective	Std. Training Objective	Mean Out-of-Sample Variance	Mean Out-of-Sample $\mathrm{CVaR}_{0.95}$	Mean Gap
30	440.62	4.84	129.82	456.37	0.00
50	444.38	6.80	129.82	456.37	0.00
100	452.25	8.41	278.37	482.71	0.1386
200	449.95	9.12	416.44	492.62	0.2609

The 30- and 50-scenario models were solved to convergence within the common computational budget and produced the same mean out-of-sample variance and $\mathrm{CVaR}_{0.95}$. This indicates that, for the present benchmark instance, increasing the training sample from 30 to 50 did not materially change the evaluated out-of-sample performance of the resulting routing solutions.

The 100- and 200-scenario settings contain more observations in the nominal upper tail, but their positive final optimality gaps indicate that the corresponding models were not fully solved within 600 s. Their higher out-of-sample variance and $\mathrm{CVaR}_{0.95}$ therefore reflect both the increase in training sample size and incomplete optimization and should not be interpreted as evidence that increasing N worsens solution stability.

Accordingly, this analysis does not establish a universal minimum sample size for accurate $\mathrm{CVaR}_{0.95}$ estimation. It shows only that, for the current instance and computational budget, the converged solutions obtained with 30 and 50 training scenarios exhibit comparable performance when evaluated on a common set of 1000 independent out-of-sample scenarios.

6.4.2. Effect of Fleet Size $|K|$

To examine the effect of fleet size on the total objective value and risk terms, three configurations, $|K| = 6, 8, 10$, are tested under the fixed setting of 16 points, 30 scenarios, $\lambda = 0.5$, and $\alpha = 0.95$.

Table 12 shows that the instances with $|K| = 8$ and $|K| = 10$ reached the prescribed convergence criterion, whereas the instance with $|K| = 6$ terminated at the 600 s time limit with a final gap of 31.34%. Thus, the $|K| = 6$ row reflects the feasible solution quality obtained under tight fleet capacity within the common time budget, and its convergence status is separately indicated in both Table 12 and Figure 6.

Table 12. Sensitivity analysis of fleet size.

$\|K\|$	Z^* (yuan)	$C_1(x)$ (yuan)	$\mathbb{E}[Q]$ (yuan)	$\text{CVaR}_{0.95}$ (yuan)	Final Gap	Status
6	484.44 *	415.94	36.07	100.93	31.34%	Time limit
8	429.38	421.93	1.11	13.78	0.00%	Converged
10	454.23	454.23	0.00	0.00	0.00%	Converged

Note: * indicates that the $|K| = 6$ instance terminated at the 600 s time limit, and the table reports the corresponding best feasible solution.

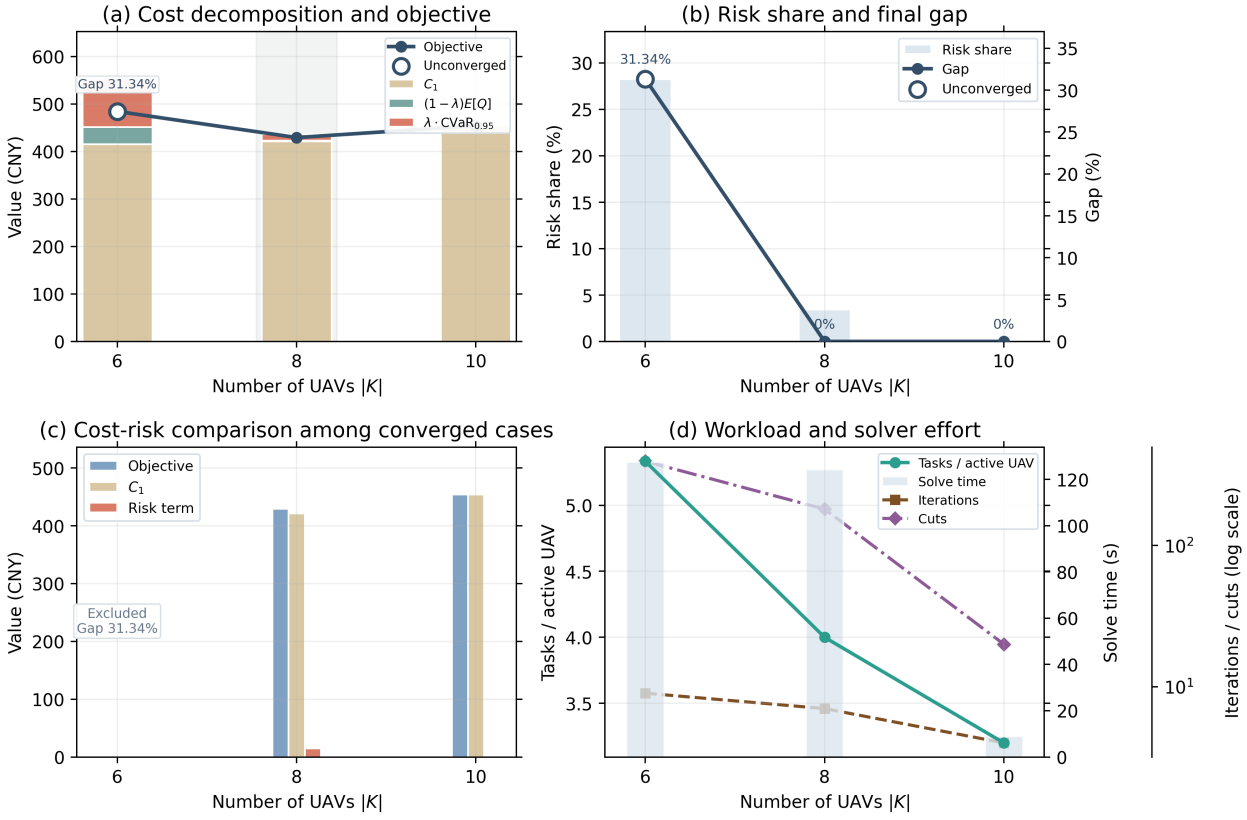


Figure 6. Sensitivity analysis of fleet size.

From a trend perspective, the feasible solution obtained for $|K| = 6$ corresponds to greater scheduling pressure and tail risk exposure under limited fleet capacity. Since this instance retains a final gap of 31.34%, the subsequent quantitative comparison focuses on the converged cases with $|K| = 8$ and $|K| = 10$: when the fleet size increases from 8 to 10, the risk term further decreases to 0, but the first-stage transportation cost increases from CNY 421.93 to CNY 454.23 and the total objective value rises from CNY 429.38 to 454.23 yuan, indicating that excessive UAV deployment can offset the benefit of risk reduction through additional transportation cost.

Figure 6 visualizes these results. The point for $|K| = 6$ is marked with a hollow marker and the label "Gap = 31.34%" to indicate its unconverged status, and Figure 6c reports the objective value, transportation cost, and risk term only for the two converged instances, $|K| = 8$ and $|K| = 10$. Figure 6a shows the cost composition and objective-value changes, Figure 6b reports the risk share and final gap, and Figure 6d illustrates task load and solver effort.

The $|K| = 6$ point corresponds to the best feasible solution obtained within the 600 s time limit, with a final gap of 31.34%, and is shown with a hollow marker.

The figure further shows that the feasible solution for $|K| = 6$ indicates possible waiting and delay accumulation under limited resources. Among the converged instances, $|K| = 8$ achieves better balance between cost and risk, whereas $|K| = 10$ further reduces the risk term but increases transportation cost. Thus, for the current instance, a moderate fleet size improves scheduling robustness while maintaining routing economy.

From the perspective of emergency rescue practice, insufficient UAV resources may place some tasks at the end of long task chains, thereby increasing tail delay risk. Excessive UAV resources, however, may increase travel and deployment costs. The resource-allocation effect is reflected in the redistribution of task loads and route sequences across the available UAVs. Under limited resources, more tasks are concentrated on longer routes, which may increase downstream waiting, delay, and tail risk exposure. Increasing the number of UAVs disperses task loads and reduces risk-related costs while also raising first-stage routing expenditure. Emergency UAV allocation needs to balance resource availability, route economy, and delay risk.

6.4.3. Effect of Confidence Level α

In addition to resource scale, the risk parameter affects how the model characterizes tail scenarios. Based on the tail distribution of a representative solution, this section examines how the number of tail scenarios, CVaR, and the objective value change under different confidence levels α, thereby illustrating the effect of risk-measurement intensity.

Table 13 shows that as α increases, CVaR places greater emphasis on higher-quantile tail scenarios, and the objective value rises accordingly. From $\alpha = 0.80$ to $\alpha = 0.99$, CVaR increases from 5.57 yuan to 20.01 yuan. At the same time, the number of scenarios classified as tail scenarios gradually decreases. The tail range considered by CVaR therefore becomes narrower, and the influence of a single extreme scenario is amplified. As a result, the risk measure may exhibit discrete jumps. Because the 30-scenario training set contains limited tail observations, these results should be interpreted together with the sample-size and out-of-sample evaluation in Section 6.4.1.

Table 13. Sensitivity analysis of α.

α	Number of Tail Scenarios	CVaR_α	Z^* (yuan)
0.80	6	5.57	425.27
0.85	5	6.68	425.83
0.90	3	11.13	428.05
0.95	2	13.78	429.38
0.99	1	20.01	432.49

This result indicates that the benefit of risk aversion depends not only on whether CVaR is introduced but also on the intensity of risk characterization. If α is too low, the model is insufficiently sensitive to tail scenarios. If α is too high, the result may be dominated by a very small number of extreme scenarios, leading to greater statistical fluctuation.

6.4.4. Effect of Risk Preference Coefficient λ

The risk preference coefficient λ determines the relative weights of the expectation and CVaR terms in the objective function. To compare the practical performance of solutions under different risk preferences, out-of-sample evaluation is conducted on the largest tested instance with 16 disaster-affected points. First-stage routes are first solved using $|\Omega_{\text{train}}| = 30$ training scenarios. The resulting fixed routes are then evaluated under $|\Omega_{\text{test}}| = 100$ independently generated out-of-sample scenarios. Four values of $\lambda = 0, 0.25, 0.5, 0.75$ are tested, with $\alpha = 0.95$ fixed.

Table 14 shows that solutions under different risk preference coefficients exhibit substantial differences in out-of-sample performance. Compared with the risk-neutral solution ($\lambda = 0.00$), increasing λ to 0.25 and 0.50 reduces the out-of-sample mean, the 95% and 99% quantiles, and $\text{CVaR}_{0.95}$. Under the current independent out-of-sample evaluation setting, $\lambda = 0.50$ gives the lowest out-of-sample mean and tail risk indicators. When λ further increases to 0.75, these indicators rise again. Moderate risk aversion therefore improves both average performance and tail performance in this test setting, whereas an overly strong risk preference may make the solution too conservative. The training-point gap for $\lambda = 0$ remains relatively high; this point is mainly used to reflect the overall difference between risk-neutral and risk-averse solutions. The other convergent points are more suitable for strict quantitative comparison.

Table 14. Sensitivity analysis of λ.

λ	Out-of-Sample Mean	Out-of-Sample Standard Deviation	95% Quantile	99% Quantile	$\text{CVaR}_{0.95}$
0.00	749.02	62.04	862.52	904.26	894.73
0.25	454.74	10.16	473.15	501.61	490.94
0.50	429.15	6.81	447.05	453.71	454.53
0.75	452.70	28.39	499.00	529.44	519.60

When λ increases from 0.50 to 0.75, $\text{CVaR}_{0.95}$ and the high-quantile indicators increase again. Under the 30-scenario training setting, $\text{CVaR}_{0.95}$ is determined by only a small number of tail scenarios. A larger λ assigns greater weight to these tail observations, making the resulting route more sensitive to the particular extreme samples included in the training set. Therefore, the inferior out-of-sample performance of $\lambda = 0.75$ may reflect both excessive conservatism and the statistical instability of tail risk estimation under a limited training sample. This also indicates that the preferred value of λ is linked to the confidence level α, the number of training scenarios, and the evaluation distribution. In the present independent out-of-sample setting, $\lambda = 0.50$ provides a more balanced choice between average performance and tail risk control.

Figure 7 shows the trends of out-of-sample indicators under different risk preference coefficients. Figure 7a–c correspond to the out-of-sample mean, the 95% quantile, and the 99% quantile together with $\text{CVaR}_{0.95}$, respectively. Figure 7d presents the training objective value and out-of-sample mean under different λ values. The out-of-sample indicators first decrease and then increase as λ increases. When λ increases from 0 to 0.50, the solution becomes less sensitive to tail disturbances; when λ reaches 0.75, the out-of-sample indicators worsen.

Overall, the risk preference coefficient affects not only the weight allocation between the expectation and risk terms in the objective function but also the actual performance of the solution under unseen scenarios. For emergency UAV rescue, moderate risk aversion helps control tail losses while maintaining overall scheduling efficiency. Excessive risk aversion may make the solution overly conservative and reduce resource utilization efficiency. The performance of solutions with a high risk preference is also jointly affected

by the representativeness of tail samples and the confidence level setting. The risk preference coefficient should therefore be determined by jointly considering task timeliness, the criticality of task chains, and the level of environmental uncertainty.

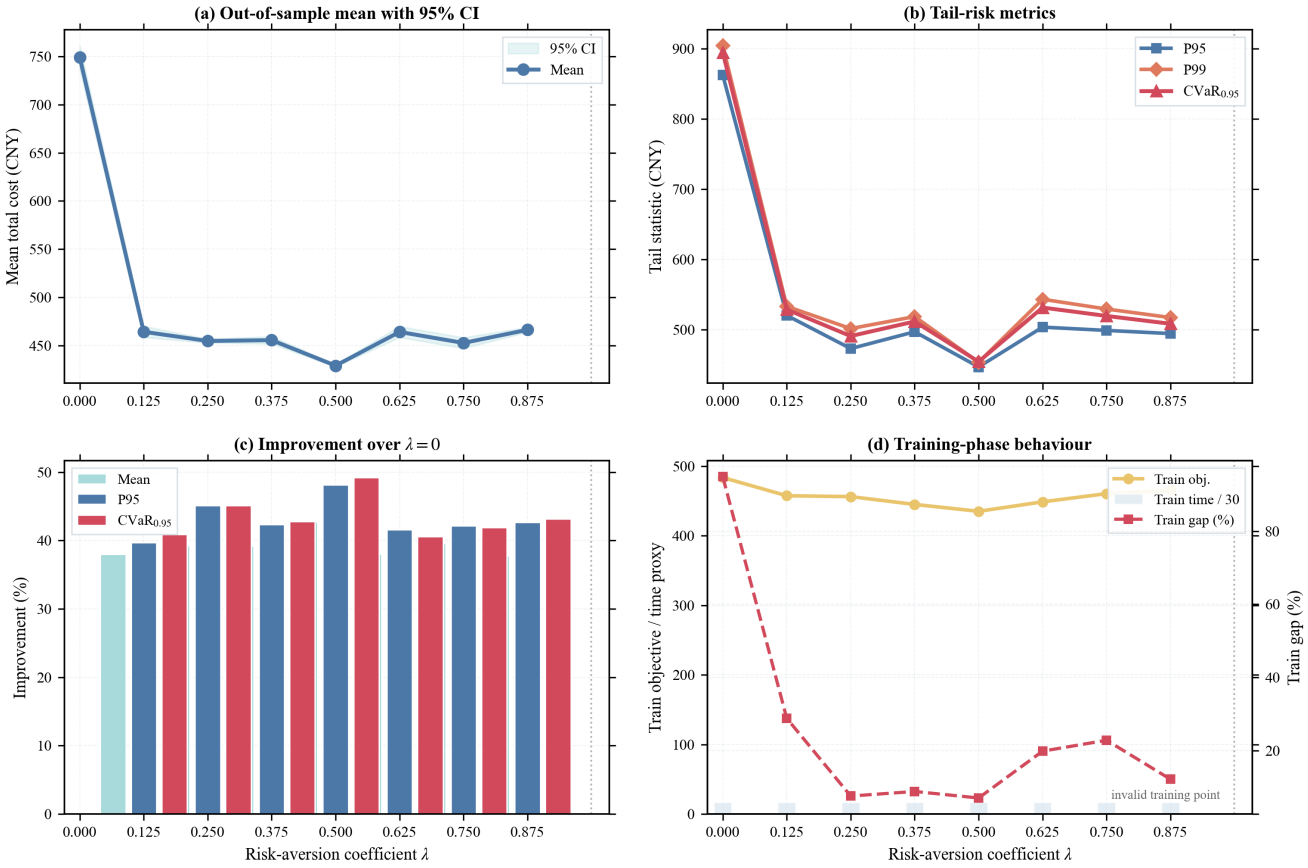


Figure 7. Sensitivity analysis of the risk preference coefficient.

6.4.5. Effect of Flight-Time and Service-Time Fluctuations

Given the resource scale and risk parameters, the fluctuation intensity of flight time and service time further affects whether the risk-averse model demonstrates an advantage. Therefore, $\alpha = 0.95$ is fixed on a representative 16-point instance, and the out-of-sample performance of the risk-neutral model ($\lambda = 0$) and the moderately risk-averse model ($\lambda = 0.5$) is compared under low, medium, and high fluctuation levels. These levels correspond to flight-time coefficients of variation of 10%/15%/25% and service-time coefficients of variation of 10%/20%/30%, respectively. The coefficient of variation is the ratio between the standard deviation and the mean of the corresponding time parameter and is used to characterize relative fluctuation in flight time and service time.

Table 15 shows that under low fluctuation, the risk-neutral and risk-averse models have similar out-of-sample mean, out-of-sample 99% quantile, and $\mathrm{CVaR}_{0.95}$. In this case, time disturbances have limited impact on scheduling outcomes, and the effect of risk-averse modeling is limited. Under medium and high fluctuation levels, the $\lambda = 0.5$ solution outperforms the risk-neutral model in both the out-of-sample 99% quantile and $\mathrm{CVaR}_{0.95}$, and the difference becomes larger under high fluctuation. The out-of-sample 99% quantile reflects the loss level in relatively adverse scenarios while $\mathrm{CVaR}_{0.95}$ reflects the average loss in the worst 5% of scenarios. These results indicate that the advantage of the risk-averse model lies mainly in reducing tail-scenario losses. When time disturbances are large, it more effectively controls costs under extreme adverse scenarios.

Table 15. Sensitivity results for flight-time and service-time fluctuations.

Fluctuation Level	λ	Out-of-Sample Mean	Out-of-Sample Std. Dev.	Out-of-Sample 99% Quantile	$CVaR_{0.95}$
Low	0.0	424.27	0.00	424.27	424.27
Low	0.5	424.27	0.00	424.27	424.27
Medium	0.0	405.34	9.67	446.07	441.98
Medium	0.5	404.16	0.00	404.16	404.16
High	0.0	583.57	59.45	746.02	732.60
High	0.5	482.81	46.36	609.48	605.71

In emergency UAV rescue, the scheduling model can be selected according to the level of time uncertainty in the task environment. When disaster conditions are relatively stable and route disturbances are small, a solution that minimizes average cost may be sufficient. When flight time and on-site service time fluctuate more substantially, a risk-averse model that jointly considers average cost and tail risk is more appropriate. Thus, the risk-averse model is mainly suitable for emergency environments with large time disturbances and potential tail losses. In practical applications, decision-makers can estimate the fluctuation level from historical task records, disaster information, and route conditions, and then decide whether to introduce CVaR-based risk control and the corresponding risk preference coefficient.

6.4.6. Sensitivity to Correlated Flight-Time Disturbances

The baseline experiments generate flight-time and service-time scenarios through independent Monte Carlo sampling. Since post-disaster weather disturbances, temporary airspace restrictions, or route-control measures may affect multiple flight arcs simultaneously, an additional sensitivity experiment is conducted under correlated flight-time disturbances. This experiment does not change the two-stage model, the Mean-CVaR objective, or the Enhanced BD solution framework. Instead, a common disturbance factor is introduced in the scenario-generation process to examine the sensitivity of the conclusions to common arc-level flight-time disturbances.

Specifically, the truncated normal marginal distribution of each flight time is retained. For arc a in scenario ω, the standardized disturbance is generated as

$$z_{a\omega} = \sqrt{\rho} G_\omega + \sqrt{1-\rho}\varepsilon_{a\omega}. \tag{29}$$

where $G_\omega \sim N(0,1)$ is the common scenario-level disturbance and $\varepsilon_{a\omega} \sim N(0,1)$ is the arc-specific local disturbance. The common disturbance and all local disturbance terms are mutually independent, and the local disturbances are independently generated across arcs. The parameter $\rho \in [0,1]$ controls the common correlation strength; under this construction, the standardized disturbances of any two distinct arcs $a \neq b$ have correlation $\mathrm{Corr}(z_{a\omega}, z_{b\omega}) = \rho$.

The standardized disturbance is then mapped back to the original truncated normal marginal distribution as

$$t_{a\omega} = F_a^{-1}(\Phi(z_{a\omega})). \tag{30}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution and $F_a^{-1}(\cdot)$ denotes the inverse cumulative distribution function of the truncated normal flight-time distribution for arc a. This construction allows different arcs in the same scenario to share a common disturbance while retaining arc-specific random variation. Service times mainly reflect task-site operational variability and are therefore kept as independent local task-level disturbances.

The experiment is conducted on the 16-point benchmark instance. To avoid mixing the effect of correlation with the effect of re-optimizing routes, two fixed routes are first obtained from independent training scenarios using the risk-neutral model ($\lambda = 0$) and the moderately risk-averse model ($\lambda = 0.5$). These 2 routes are then evaluated on 1000 out-of-sample scenarios generated from the same base random numbers under $\rho = 0, 0.3, 0.6$, with $\alpha = 0.95$. Table 16 reports the out-of-sample mean, standard deviation, 95% quantile, 99% quantile, and $\text{CVaR}_{0.95}$.

Table 16 shows that, when the routes and marginal distributions are kept fixed, stronger flight-time correlation mainly amplifies out-of-sample variability and tail losses. For the route trained with $\lambda = 0$, increasing ρ from 0 to 0.6 raises the out-of-sample mean only moderately from 433.73 to 445.05 but increases the out-of-sample standard deviation from 19.10 to 47.75, the 95% quantile from 470.50 to 551.43, and $\text{CVaR}_{0.95}$ from 493.65 to 612.95. The route trained with $\lambda = 0.5$ shows a similar pattern. This indicates that common flight-time disturbances mainly affect performance under adverse scenarios rather than only the average cost.

Table 16. Sensitivity results under correlated flight-time disturbances.

ρ	Training λ	Out-of-Sample Mean	Out-of-Sample Std. Dev.	95% Quantile	99% Quantile	$\text{CVaR}_{0.95}$
0.0	0.0	433.73	19.10	470.50	503.28	493.65
0.0	0.5	458.06	23.92	500.58	518.28	520.92
0.3	0.0	439.69	33.15	505.93	570.87	551.32
0.3	0.5	463.80	36.13	535.29	593.81	573.23
0.6	0.0	445.05	47.75	551.43	639.86	612.95
0.6	0.5	468.43	47.05	565.66	641.52	616.23

Under the present fixed-route and out-of-sample evaluation setting, the route trained with $\lambda = 0.5$ does not outperform the route trained with $\lambda = 0$ in terms of the reported indicators. This result indicates that a risk-averse route trained under independent scenarios does not necessarily retain a tail risk advantage when evaluated under out-of-sample scenarios with different dependence structures. Therefore, the risk preference coefficient should be selected together with out-of-sample validation that reflects the dependence characteristics of the intended operating environment.

Overall, the common-factor experiment shows that the two-stage model and the Enhanced BD framework can directly accept discrete scenarios with correlated flight times. Arc-level independence is therefore an assumption of the baseline scenario generator rather than a structural requirement of the optimization model. At the same time, correlation substantially changes the out-of-sample tail loss distribution, suggesting that independent scenario sampling may underestimate systemic tail risk when weather or airspace disruptions affect multiple flight arcs simultaneously. The common factor structure provides a concise scenario-generation scheme for representing regional common disturbances and offers a basis for incorporating more detailed spatiotemporal dependence structures in future extensions.

6.5. Generalization Across Different Network Topologies

To examine whether the above findings depend on the single geographic structure of the Jiangning instance, three synthetic network topologies are further constructed: clustered, dispersed, and corridor-shaped networks. Each synthetic instance contains 16 disaster-affected points, 32 reconnaissance and delivery task nodes, and 8 UAVs. The number of training scenarios is 30, and the out-of-sample evaluation uses 100 independent test scenarios. Flight speed, service times, deadline-generation rules, uncertainty distributions, cost parameters, and risk parameters are kept consistent with the Jiangning benchmark.

Therefore, the main difference among these instances lies in the spatial distribution of the disaster-affected points.

In the clustered topology, disaster-affected points are generated around three local centers, representing a situation in which rescue demand is concentrated in several affected areas. In the dispersed topology, points are generated approximately uniformly over the rectangular study region, representing a spatially scattered demand pattern. In the corridor topology, points are distributed along a main spatial axis, representing emergency tasks that occur along a road, river, or elongated disaster belt. Figure 8 shows the spatial structures of the Jiangning instance and the three synthetic topologies.

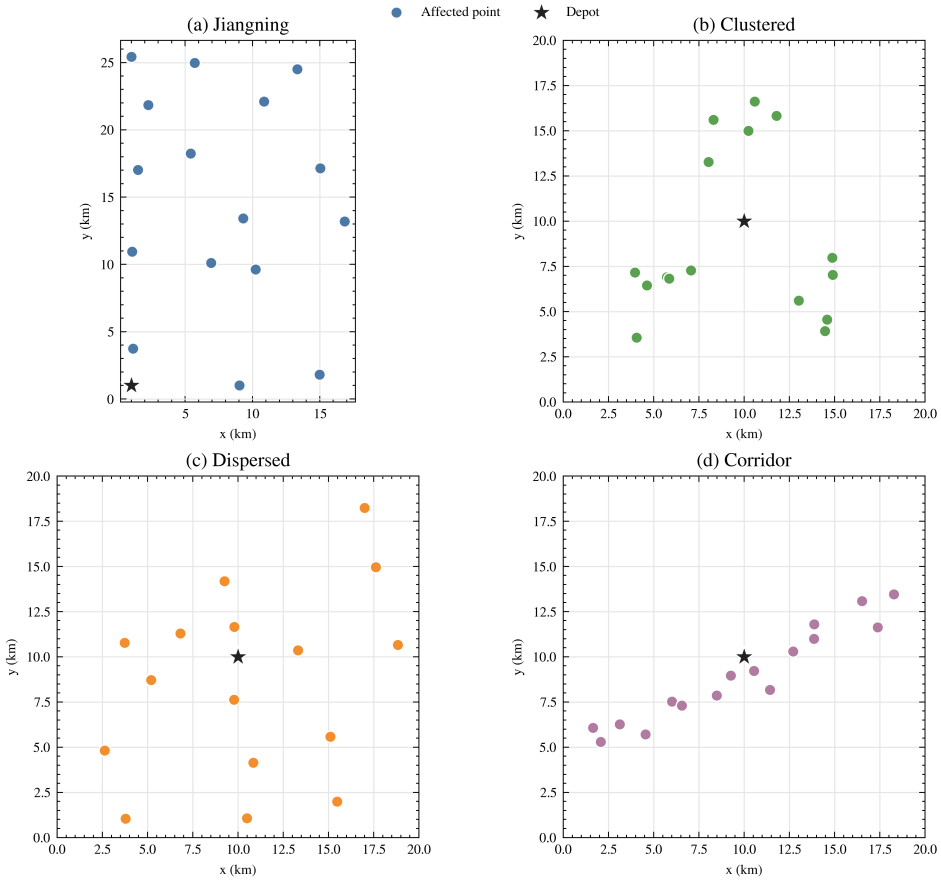


Figure 8. Spatial distributions under different network topologies.

For each synthetic topology, the risk-neutral model ($\lambda = 0$) and the moderately risk-averse model ($\lambda = 0.5$) are solved. The resulting first-stage schedules are then evaluated on the same set of 100 independent out-of-sample scenarios. Table 17 reports the out-of-sample mean, standard deviation, 99% quantile, and $\mathrm{CVaR}_{0.95}$. The runtime corresponds to the Enhanced BD solution time for the corresponding training problem.

Table 17. Out-of-sample performance under different synthetic topologies.

Topology	λ	Out-of-Sample Mean	Out-of-Sample Std. Dev.	Out-of-Sample 99% Quantile	$\mathrm{CVaR}_{0.95}$	Runtime (s)
Clustered	0.0	467.12	57.25	602.14	583.48	171.2
Clustered	0.5	259.28	6.84	283.15	282.84	170.8
Dispersed	0.0	535.59	59.99	657.53	647.80	515.8
Dispersed	0.5	334.01	24.42	391.82	388.51	514.7
Corridor	0.0	209.42	25.38	271.49	271.95	513.6
Corridor	0.5	207.64	12.04	249.74	244.82	515.0

Table 17 and Figure 9 show that network topology substantially affects the absolute scheduling cost, the benefit of risk aversion, and the computational difficulty. Compared with the risk-neutral solution, the $\lambda = 0.5$ solution reduces the out-of-sample mean, 99% quantile, and $CVaR_{0.95}$ by 44.5%, 53.0%, and 51.5%, respectively, in the clustered topology. In the dispersed topology, the corresponding reductions are 37.6%, 40.4%, and 40.0%. These results indicate that, when demand points are locally clustered or widely dispersed, the Mean-CVaR objective encourages routing structures that are less sensitive to time disturbances. As a result, it improves not only tail risk performance but also the out-of-sample average performance under the finite training scenario setting.

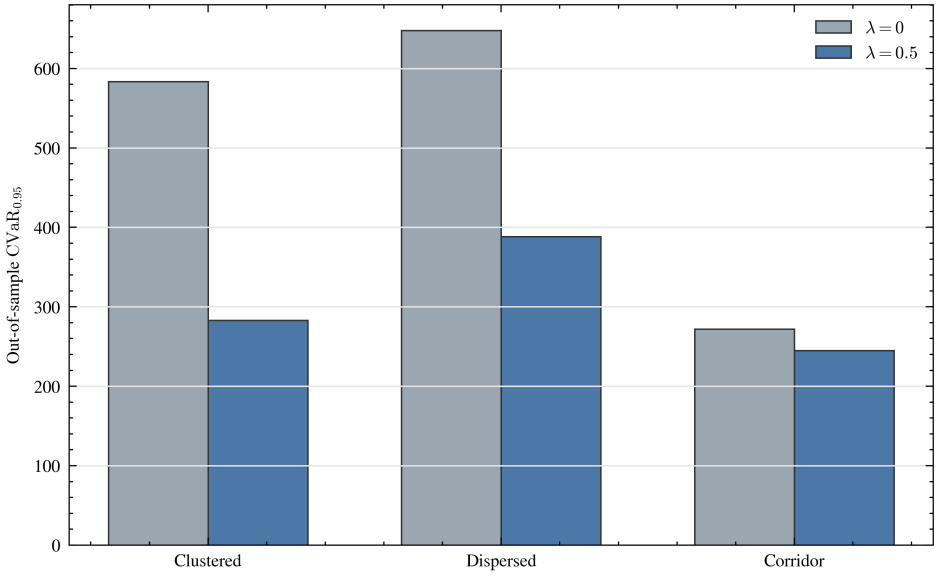


Figure 9. Comparison of out-of-sample CVaR under different synthetic topologies.

The corridor topology shows a different pattern. Compared with the risk-neutral solution, the $\lambda = 0.5$ solution reduces the out-of-sample mean by only 0.8%, but still reduces the standard deviation, 99% quantile, and $CVaR_{0.95}$ by 52.6%, 8.0%, and 10.0%, respectively. This is because the affected points in the corridor topology are distributed along a single main axis, which limits the routing alternatives and gives the risk-neutral solution a relatively regular visiting order. Therefore, risk aversion has less room to improve the average cost, but it still reduces performance variability and extreme-scenario losses.

The runtimes of the $\lambda = 0$ and $\lambda = 0.5$ solutions are similar within the same topology, indicating that introducing the Mean-CVaR objective does not noticeably increase the computational burden of Enhanced BD in these tests. By contrast, runtime differs substantially across topologies: the clustered instance is solved in about 171 s, whereas the dispersed and corridor instances both require more than 510 s. This suggests that, even when the number of tasks, UAVs, and scenarios remains unchanged, the spatial distribution of nodes and the induced routing structure can strongly affect computational difficulty. Overall, the risk-averse solution reduces the out-of-sample 99% quantile and $CVaR_{0.95}$ in all three synthetic topologies, indicating that the tail risk control effect of the proposed model is not driven solely by the Jiangning geographic structure. However, the magnitude of improvement and the runtime differ across topologies, showing that network structure remains an important factor affecting both risk-averse benefit and computational performance. Therefore, this experiment supports the robustness of the proposed method under different spatial distribution patterns, while its cross-regional applicability still requires further validation using additional real-world regional data.

7. Conclusions and Future Work

This study develops a risk-averse two-stage stochastic programming model for emergency UAV task allocation under uncertain flight and service times. In the first stage, the model determines task assignment, routing, and execution order. In the second stage, it evaluates waiting, delay, and recourse costs under stochastic scenarios. The Mean-CVaR risk measure is used to describe tail delay losses in extreme scenarios. Compared with an expected cost-based scheduling model, the proposed model captures both average operational performance and tail risk exposure, and is therefore suitable for emergency rescue tasks with reconnaissance–delivery precedence relations and time disturbances.

To solve the model, this study develops a problem-tailored Enhanced BD computational framework by integrating partial Benders, heuristic warm start, and dynamic cut-pool management into the classical Benders structure. Ten independent replications show that Enhanced BD reaches convergence in all ten medium-sized replications, whereas Basic BD does not converge in any replication. In the largest tested instance, Enhanced BD reaches the convergence criterion in eight out of ten replications, while DEP and Basic BD do not converge. The extended cross-scale ablation experiment further shows that Partial Benders is the most influential single acceleration component in the tested instances. It closes the BD gap in the medium-sized instance and yields the lowest final gap among the single-strategy configurations in the largest tested instance. However, because partial Benders alone does not fully close the gap in the largest tested instance, the full Enhanced BD framework remains necessary for robust convergence. These results indicate that the proposed framework reduces the computational burden of the multi-scenario risk-averse model and improves solution stability within the tested instance range.

Further analysis shows that tail delay risk is concentrated at a small number of demand points and downstream nodes in their task chains. This indicates that time disturbances from preceding tasks can propagate along the task chain and accumulate at local downstream nodes. The sensitivity analysis shows that, under the independent out-of-sample evaluation setting, moderate risk aversion improves out-of-sample performance, with $\lambda = 0.50$ performing well in that test setting. Excessive risk aversion may make the solution overly conservative and reduce scheduling efficiency. At the same time, the out-of-sample performance under a higher risk preference may also be affected by the representativeness of tail samples and the choice of α. The correlated flight-time sensitivity analysis further shows that the proposed model and algorithm can accept discrete scenarios with a correlated structure. Under fixed routes and approximately consistent marginal distributions, common arc-level disturbances substantially increase the out-of-sample standard deviation, high-quantile losses, and $\mathrm{CVaR}_{0.95}$, indicating that independent scenario sampling may underestimate systemic tail risk. The fleet size analysis indicates that tighter UAV resources are associated with greater scheduling pressure and tail risk exposure; among the converged instances, a moderate fleet size provides a better balance between risk control and transportation cost. Further increasing the number of UAVs can reduce risk-related costs, but the additional transportation cost weakens the overall benefit. The additional synthetic topology experiments further show that the Mean-CVaR risk-averse solution reduces high-quantile out-of-sample losses and $\mathrm{CVaR}_{0.95}$ in clustered, dispersed, and corridor-shaped networks, indicating that the main findings are not limited to the single Jiangning topology.

This study still has several limitations. It mainly considers flight-time and service-time disturbances. The current model represents part of the influence of environmental disturbances through stochastic flight time and service time, but it does not explicitly model disaster-state or on-site environmental changes as dynamic state variables. Some effects of temporary detours, short-term airspace restrictions, and information transmission

delays can be indirectly reflected through variations in flight time or service time. However, sudden demand changes, UAV failures, complete route closures, severe communication disruptions, dynamic obstacles, inter-UAV flight conflicts, and airspace capacity restrictions may alter the task set, available UAV fleet, network connectivity, or operational feasibility and are not explicitly represented in the current model. From a qualitative sensitivity perspective, demand fluctuations may change task priorities and payload requirements, UAV failures may reduce available service capacity, route closures and dynamic obstacles may increase travel detours or make some paths infeasible, severe communication disruptions may delay information updates and task adjustment, and inter-UAV conflicts or limited airspace capacity may restrict the simultaneous use of flight routes. In actual rescue operations, changing weather, route conditions, obstacle distribution, communication status, and disaster severity may further alter task urgency, route suitability, and the feasibility of the original scheduling plan. These factors may amplify waiting and delay costs beyond the level captured by flight-time and service-time variations, especially under extreme post-disaster conditions. Future research may incorporate scenario-dependent demand, route and UAV availability, dynamic route adjustment, task-reallocation mechanisms, inter-UAV conflict resolution, and airspace capacity constraints to address these structural and operational uncertainties. Future research may also incorporate real-time environmental information and task urgency into rolling horizon or multistage scheduling models to investigate dynamic task priority updating and corresponding UAV route rescheduling.

The present study focuses on static pre-planning for UAV reconnaissance and emergency-material delivery. The computational comparison is conducted among model-consistent mathematical programming and Benders-based methods. Future research may construct a unified dynamic decision environment to compare the proposed framework with reinforcement learning or rolling horizon scheduling methods under online task arrivals and real-time operational disturbances. Meanwhile, the proposed model focuses on risk-averse pre-planning under predetermined task assignment and routing decisions, and the second stage mainly describes the execution outcome of a fixed routing plan under time disturbance scenarios. Route diversion, task cancellation, and cross-UAV reassignment involve scenario-dependent route recovery decisions and belong to dynamic recovery and online rescheduling problems, which will be further investigated in future research. The current model does not further distinguish delivery preparation, pre-positioning, and formal service stages. Future research may study soft-precedence relations and real-time coordination mechanisms when partial reconnaissance information is available before the reconnaissance task is fully completed.

In addition, the current model adopts a static endurance constraint based on total flight time and does not explicitly account for the effects of task payload variations, flight states, and nonlinear battery discharge characteristics on energy consumption. Therefore, the model is mainly applicable to scheduling problems involving lightweight emergency deliveries and may overestimate the energy feasibility of some routes under heavy-load or complex flight conditions. Future research may introduce payload state, arc-level energy consumption, and residual battery variables to couple energy consumption with routing decisions and task execution processes.

Although the instances are constructed from real geographic and demand-point data, the stochastic disturbance scenarios and scheduling process are still generated mainly through simulation. Further validation under conditions closer to actual UAV operations is needed. In addition, this study uses a common factor structure to generate correlated flight-time scenarios and analyze the effect of common weather or airspace disturbances on tail risk. Future research may further combine real weather, airspace-control, or communication time-series data with spatiotemporal random fields, copula-based dependence

structures, or data-driven scenario generation methods to represent correlated disruptions in post-disaster environments in greater detail. Although the computational experiments gradually increase the numbers of task nodes, UAVs, and stochastic scenarios, the largest tested configuration contains only 16 disaster-affected points, corresponding to 32 task nodes, 8 UAVs, and 100 scenarios. This scale remains limited relative to standard large-scale vehicle routing benchmarks. Therefore, the reported results primarily reflect the computational performance of the proposed framework within the tested instance range. Future research will evaluate the proposed method on substantially larger task networks and scenario sets to further examine its scalability.

Future research may incorporate dynamic task arrivals and real-time disaster information into rolling horizon or multistage risk-averse scheduling models so that task allocation plans can be updated as the environment changes. Demand variation, temporary no-fly zones, communication interruptions, and equipment failures can also be included to improve the model's applicability to complex post-disaster environments. Future research may further distinguish ground-idle waiting from aerial hovering and incorporate hovering energy consumption into the endurance constraint, thereby improving the applicability of the model to operations in which temporary landing is unavailable. In addition, the proposed framework can be extended to heterogeneous UAV systems and collaborative rescue systems involving multiple resource types. Future research may also investigate the integration of scenario reduction or progressive scenario decomposition techniques with the proposed decomposition framework to further improve computational scalability while controlling the approximation error introduced by scenario selection, probability redistribution, or scenario coordination strategies. Moreover, larger task sets, multi-center emergency networks, real disaster data, and semi-physical simulation platforms may be considered to further evaluate the practical performance of the Mean-CVaR model and the Enhanced BD solution framework.

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

Funding: This research was funded by the Natural Science Foundation of Jiangsu Province, grant number BK20253036.

Data Availability Statement: The data supporting this study are available from the corresponding author upon reasonable request. The coordinates were derived from the authors' previous study, and the stochastic scenarios were generated for this paper.

Acknowledgments: The authors sincerely thank the editors and reviewers for their valuable comments and suggestions on this manuscript.

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

References

1. Carnero Quispe, C.M.; Couto, A.S.; de Brito Junior, I.; Cunha, L.R.A.; Siqueira, R.M.; Yoshizaki, H.T.Y. Humanitarian Logistics Prioritization Models: A Systematic Literature Review. *Logistics* **2024**, *8*, 60. [CrossRef]
2. Wang, Y.; Li, J.; Yang, X.; Peng, Q. UAV-Ground Vehicle Collaborative Delivery in Emergency Response: A Review of Key Technologies and Future Trends. *Appl. Sci.* **2025**, *15*, 9803. [CrossRef]
3. Liu, H.; Shao, Z.; Zhou, Q.; Tu, J.; Zhu, S. Task Allocation Algorithm for Heterogeneous UAV Swarm with Temporal Task Chains. *Drones* **2025**, *9*, 574. [CrossRef]
4. Yin, Y.; Yang, Y.; Yu, Y.; Wang, D.; Cheng, T.C.E. Robust vehicle routing with drones under uncertain demands and truck travel times in humanitarian logistics. *Transp. Res. Part B Methodol.* **2023**, *174*, 102781. [CrossRef]

5. Graß, E.; Fischer, K. Two-stage stochastic programming in disaster management: A literature survey. *Surv. Oper. Res. Manag. Sci.* **2016**, *21*, 85–100. [CrossRef]
6. Miller, N.; Ruszczyński, A. Risk-Averse Two-Stage Stochastic Linear Programming: Modeling and Decomposition. *Oper. Res.* **2011**, *59*, 125–132. [CrossRef]
7. Rockafellar, R.T.; Uryasev, S. Optimization of conditional value-at-risk. *J. Risk* **2000**, *2*, 21–41. [CrossRef]
8. Noyan, N. Risk-averse two-stage stochastic programming with an application to disaster management. *Comput. Oper. Res.* **2012**, *39*, 541–559. [CrossRef]
9. Torres, J.J.; Li, C.; Apap, R.M.; Grossmann, I.E. A Review on the Performance of Linear and Mixed Integer Two-Stage Stochastic Programming Software. *Algorithms* **2022**, *15*, 103. [CrossRef]
10. Rahmaniani, R.; Crainic, T.G.; Gendreau, M.; Rei, W. The Benders decomposition algorithm: A literature review. *Eur. J. Oper. Res.* **2017**, *259*, 801–817. [CrossRef]
11. Lyu, M.; Zhao, Y.; Huang, C.; Huang, H. Unmanned Aerial Vehicles for Search and Rescue: A Survey. *Remote Sens.* **2023**, *15*, 3266. [CrossRef]
12. Tang, M.; Hu, M.; Zhang, H.; Zhou, L. Research on Multi Unmanned Aerial Vehicles Emergency Task Planning Method Based on Discrete Multi-Objective TLBO Algorithm. *Sustainability* **2022**, *14*, 2555. [CrossRef]
13. Rottondi, C.; Malandrino, F.; Bianco, A.; Chiasserini, C.F.; Stavrakakis, I. Scheduling of emergency tasks for multiservice UAVs in post-disaster scenarios. *Comput. Netw.* **2021**, *184*, 107644. [CrossRef]
14. Chowdhury, S.; Shahvari, O.; Marufuzzaman, M.; Li, X.; Bian, L. Drone routing and optimization for post-disaster inspection. *Comput. Ind. Eng.* **2021**, *159*, 107495. [CrossRef]
15. Lin, Z.; Xu, X.; Demir, E.; Laporte, G. Optimizing task assignment and routing operations with a heterogeneous fleet of unmanned aerial vehicles for emergency healthcare services. *Comput. Oper. Res.* **2025**, *174*, 106890. [CrossRef]
16. Gao, Z.; Zheng, M.; Zhong, H.; Mei, Y. Robust Optimization for Cooperative Task Assignment of Heterogeneous Unmanned Aerial Vehicles with Time Window Constraints. *Axioms* **2025**, *14*, 184. [CrossRef]
17. Huang, H.; Jiang, Z.; Yan, T.; Bai, Y. Dynamic Task Allocation for Heterogeneous Multi-UAVs in Uncertain Environments Based on 4DI-GWO Algorithm. *Drones* **2024**, *8*, 236. [CrossRef]
18. Ghelichi, Z.; Gentili, M.; Mirchandani, P.B. Drone logistics for uncertain demand of disaster-impacted populations. *Transp. Res. Part C Emerg. Technol.* **2022**, *141*, 103735. [CrossRef]
19. Nunes, E.; McIntire, M.; Gini, M. Decentralized multi-robot allocation of tasks with temporal and precedence constraints. *Adv. Robot.* **2017**, *31*, 1193–1207. [CrossRef]
20. Ferreira, B.A.; Petrović, T.; Orsag, M.; Martínez-de-Dios, J.R.; Bogdan, S. Distributed allocation and scheduling of tasks with cross-schedule dependencies for heterogeneous multi-robot teams. *Robot. Auton. Syst.* **2022**, *150*, 103999. [CrossRef]
21. Cui, W.; Li, R.; Feng, Y.; Yang, Y. Distributed Task Allocation for a Multi-UAV System with Time Window Constraints. *Drones* **2022**, *6*, 226. [CrossRef]
22. Wang, J.; Jia, G.; Guo, Z.; Hou, Z. Impact of Uncertain Flight Time on Heterogeneous UAVs' Task Planning with Temporal Constraints. *Aerospace* **2024**, *11*, 214. [CrossRef]
23. Jeong, B.M.; Oh, Y.S.; Jang, D.S.; Hwang, N.E.; Kim, J.W.; Choi, H.L. Makespan-Minimizing Heterogeneous Task Allocation under Temporal Constraints. *Aerospace* **2023**, *10*, 1032. [CrossRef]
24. Dukkanci, O.; Koberstein, A.; Kara, B.Y. Drones for relief logistics under uncertainty after an earthquake. *Eur. J. Oper. Res.* **2023**, *310*, 117–132. [CrossRef]
25. He, Y.; Zheng, Z.; Li, H.; Deng, J. A Stochastic Drone-Scheduling Problem with Uncertain Energy Consumption. *Drones* **2024**, *8*, 430. [CrossRef]
26. Venkatachalam, S.; Sundar, K.; Rathinam, S. A Two-Stage Approach for Routing Multiple Unmanned Aerial Vehicles with Stochastic Fuel Consumption. *Sensors* **2018**, *18*, 3756. [CrossRef] [PubMed]
27. Yan, R.; Tian, H.; Gao, K.; Peng, R.; Liu, B. A two-stage UAV routing problem with time window considering rescheduling with random delivery reliability. *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.* **2023**, *237*, 781–797. [CrossRef]
28. Fábián, C.I. Handling CVaR objectives and constraints in two-stage stochastic models. *Eur. J. Oper. Res.* **2008**, *191*, 888–911. [CrossRef]
29. Ling, A.; Sun, J.; Xiu, N.; Yang, X. Robust two-stage stochastic linear optimization with risk aversion. *Eur. J. Oper. Res.* **2017**, *256*, 215–229. [CrossRef]
30. Wang, W.; Yang, K.; Yang, L.; Gao, Z. Two-stage distributionally robust programming based on worst-case mean-CVaR criterion and application to disaster relief management. *Transp. Res. Part E Logist. Transp. Rev.* **2021**, *149*, 102332. [CrossRef]
31. Gu, Y.; Huang, Y.; Wang, Y. Data-Driven Distributionally Robust Risk-Averse Two-Stage Stochastic Linear Programming over Wasserstein Ball. *J. Optim. Theory Appl.* **2024**, *200*, 242–279. [CrossRef]
32. Ghaffarinasab, N.; Çavuş, Ö.; Kara, B.Y. A mean-CVaR approach to the risk-averse single allocation hub location problem with flow-dependent economies of scale. *Transp. Res. Part B Methodol.* **2023**, *167*, 32–53. [CrossRef]

33. Khatami, M.; Silva, T.; Pagnoncelli, B.K.; Ntaimo, L. Risk-averse multistage stochastic programs with expected conditional risk measures. *Comput. Oper. Res.* **2024**, *172*, 106802. [CrossRef]
34. Benders, J.F. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* **1962**, *4*, 238–252. [CrossRef]
35. Birge, J.R.; Louveaux, F.V. A multicut algorithm for two-stage stochastic linear programs. *Eur. J. Oper. Res.* **1988**, *34*, 384–392. [CrossRef]
36. Laporte, G.; Louveaux, F.V. The integer L-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.* **1993**, *13*, 133–142. [CrossRef]
37. Huang, Y.; Zheng, Q.P.; Wang, J. Two-stage stochastic unit commitment model including non-generation resources with conditional value-at-risk constraints. *Electr. Power Syst. Res.* **2014**, *116*, 427–438. [CrossRef]
38. Taghizadeh, E.; Venkatachalam, S. Two-stage risk-averse stochastic programming approach for multi-item single source ordering problem: CVaR minimisation with transportation cost. *Int. J. Prod. Res.* **2023**, *61*, 2129–2146. [CrossRef]
39. Zhang, C.; Ng, K.K.; Jin, Z.; Sun, X.; Qin, Y. Risk-averse two-stage stochastic programming model for gate assignment problem under arrival time uncertainty. *Comput. Ind. Eng.* **2025**, *207*, 111269. [CrossRef]
40. Magnanti, T.L.; Wong, R.T. Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. *Oper. Res.* **1981**, *29*, 464–484. [CrossRef]
41. Crainic, T.G.; Hewitt, M.; Maggioni, F.; Rei, W. Partial Benders Decomposition: General Methodology and Application to Stochastic Network Design. *Transp. Sci.* **2021**, *55*, 414–435. [CrossRef]
42. Mahéo, A.; Belieres, S.; Adulyasak, Y.; Cordeau, J.F. Unified Branch-and-Benders-Cut for two-stage stochastic mixed-integer programs. *Comput. Oper. Res.* **2024**, *164*, 106526. [CrossRef]
43. Ramírez-Pico, C.; Ljubić, I.; Moreno, E. Benders Adaptive-Cuts Method for Two-Stage Stochastic Programs. *Transp. Sci.* **2023**, *57*, 1252–1275. [CrossRef]
44. Baena, D.; Castro, J.; Frangioni, A. Stabilized Benders Methods for Large-Scale Combinatorial Optimization, with Application to Data Privacy. *Manag. Sci.* **2020**, *66*, 3051–3068. [CrossRef]
45. Zhang, H.; Mazzi, N.; McKinnon, K.; Nava, R.G.; Tomasgard, A. A stabilised Benders decomposition with adaptive oracles for large-scale stochastic programming with short-term and long-term uncertainty. *Comput. Oper. Res.* **2024**, *167*, 106665. [CrossRef]
46. Heitsch, H.; Römisch, W. A note on scenario reduction for two-stage stochastic programs. *Oper. Res. Lett.* **2007**, *35*, 731–738. [CrossRef]
47. Luo, F.; Dey, S.; Mehrotra, S. A cutting-plane and Benders' decomposition algorithm for two-stage distributionally robust convex programs. *Math. Program.* **2025**, *advance online publication*. [CrossRef]
48. Zhao, S.; Yang, H.; Wang, Y.; Yang, Z. A risk-averse two-stage stochastic programming model for vessel schedule recovery in liner shipping service. *Transp. Res. Part E Logist. Transp. Rev.* **2026**, *208*, 104655. [CrossRef]
49. Wan, L.; Xu, S.; Huang, J.; Zhang, Q.; Yuan, Z. Research on UAV emergency distribution center location based on fuzzy maximal covering model. *Transp. Syst. Eng. Inf.* **2026**, *26*, 166–175. Available online: https://link.cnki.net/urlid/11.4520.U.20260508.1829.002 (accessed on 10 July 2026).

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.