

Article

UAV–Rider Collaborative Dispatching Under Stochastic Wind Conditions Considering Nonlinear Energy Dynamics

Chunxia Shangguan , Churan Zhang and Shouqi Cao *

College of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China; cxshangguan@shou.edu.cn (C.S.); m230851610@st.shou.edu.cn (C.Z.)

* Correspondence: sqcao@shou.edu.cn

Highlights

What are the main findings?

- A stochastic UAV–rider collaborative dispatching framework is developed to address delivery uncertainty under varying wind conditions.
- A nonlinear aerodynamic energy model and an improved branch-and-price algorithm enable reliable and scalable route optimization.

What is the implication of the main findings?

- The proposed approach enhances mission feasibility and robustness for last-mile delivery in windy urban environments.
- The modeling framework provides a transferable foundation for energy-aware and uncertainty-resilient drone logistics systems.

Abstract

To mitigate UAV (unmanned aerial vehicle) range limitation risks and scheduling disruptions caused by complex wind fields in urban instant delivery, this paper proposes a UAV–rider collaborative dispatching framework. By incorporating aerodynamic-based nonlinear energy dynamics, the model accurately characterizes power variations under stochastic wind conditions, significantly enhancing the operational reliability of urban delivery missions. First, an aerodynamic-based nonlinear energy function is constructed, coupling payload, airspeed, and random wind vectors to accurately characterize power variations. Second, a scenario-based two-stage stochastic programming framework is adopted, where the rider’s deterministic path is optimized in the first-stage decision to ensure stability, and the UAV’s scenario-dependent flight plan is resolved in the second stage to adapt to wind uncertainty. An improved branch-and-price (IBP) algorithm is designed to solve this large-scale model, where nonlinear energy is evaluated during label extension in the pricing sub-problem, effectively avoiding linearization errors. The numerical results demonstrate that the proposed framework improves the mission success probability (the likelihood of completing delivery routes without battery exhaustion across all considered wind scenarios) by 25% under strong-wind conditions by effectively avoiding power failure risks. Furthermore, the IBP algorithm outperforms traditional exact solvers by over 40% in solution efficiency for large-scale cases. These findings demonstrate that energy-aware stochastic dispatching significantly improves the reliability and robustness of UAV-assisted last-mile delivery in windy urban environments, thereby providing an effective operational solution for real-world drone delivery logistics.



Academic Editor: Mumtaz Karatas

Received: 16 January 2026

Revised: 28 February 2026

Accepted: 1 March 2026

Published: 4 March 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](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: integrated transportation; instant delivery; UAV-Rider collaboration; stochastic wind conditions; nonlinear energy consumption; two-stage stochastic programming; branch-and-price algorithm

1. Introduction

With the rapid expansion of instant delivery platforms, urban logistics systems are playing an increasingly vital role in ensuring public services and healthcare accessibility. In particular, in scenarios involving home-based patients or medical emergencies, the timeliness and reliability of emergency medicine delivery are directly linked to therapeutic outcomes and patient safety. In time-critical tasks such as emergency medical logistics (e.g., the rapid delivery of blood products or Automated External Defibrillators), flight failures due to unforeseen energy depletion can have life-critical consequences. This necessitates a robust dispatching framework that prioritizes reliability under environmental uncertainty, justifying our focus on this specialized domain.

However, traditional last-mile delivery modes—primarily reliant on ground riders or vehicles—often suffer from traffic congestion, rising labor costs, and rigid time-window constraints during peak periods. These challenges make it difficult to simultaneously achieve high delivery efficiency and service quality within highly uncertain urban environments. Consequently, developing an efficient, reliable, and resilient instant delivery system for emergency medicine in complex urban environments has become a critical challenge for both urban logistics and emergency management.

In recent years, unmanned aerial vehicles (UAVs) have emerged as a promising technological solution for improving due to their high mobility, flexibility, and autonomous capabilities [1]. Integrating UAVs into delivery systems and establishing a collaborative mechanism with ground riders has been widely recognized as an effective strategy for enhancing the efficiency, coverage, and resilience of last-mile logistics networks [2].

Against this backdrop, this paper explores a UAV–rider collaborative dispatching framework tailored for emergency medicine delivery. This hybrid model allows UAVs to deliver rapid aerial transport across long distances or congestion-prone areas, while riders handle flexible ground distribution and precise doorstep service. By leveraging the complementary advantages of aerial and ground delivery, such collaboration significantly boosts operational efficiency, enhances service reliability, and reduces energy consumption in complex urban settings.

To address environmental uncertainty—especially wind conditions impacting UAV operations—a scenario-based two-stage stochastic programming approach is adopted to model the collaborative dispatching process. In the first stage, rider routes and task assignments are finalized as immediate decisions. In the second stage, UAV flight paths are adaptively optimized based on actual wind scenarios to ensure energy feasibility and successful service completion [3].

2. Related Work

In this section, we review recent literature on UAV collaborative last-mile delivery and emergency medical response. At the end of this section, the differences between our plan and previous work are presented.

In the domain of emergency material distribution, Bai et al. [4] developed an intelligent drug delivery system tailored to the low-altitude economy by embedding a stochastic multi-batch non-repetitive exchange strategy into the simulated annealing algorithm. Li et al. [5] formulated a multi-objective mixed-integer nonlinear programming model by integrating

the flow of medical supplies between nodes with appointment allocation rules. Furthermore, a UAV-mediated directional first-aid delivery system developed by Sheng et al. [6] validated the application potential of UAVs in emergency medical services.

To enhance delivery coverage and operational stability, researchers have shifted their focus from single-mode UAV or vehicle delivery to air-ground collaborative modes. In this field, Long et al. [7] established a truck-UAV collaborative dispatching model for urban emergency management and developed an integrated scheduling algorithm based on tabu search. For truck-UAV routing problems with time windows, Yin et al. [8] designed a branch-and-price-and-cut algorithm to handle the combinatorial complexity of parallel UAV scheduling. Liu et al. [9] proposed a joint “UAV-vehicle” delivery model and verified its effectiveness using a simulated annealing algorithm with terminal optimization. With the rising demand for instant delivery, the UAV-rider collaborative mode has gained increasing attention. Studies indicate that introducing rider-UAV collaboration in food and instant delivery scenarios can effectively reduce operating costs and alleviate rider labor intensity [10]. Niu et al. [11] built a UAV-rider collaborative dispatching model to enhance overall system efficiency through task assignment and path optimization. Chen et al. [12] further investigated the influence of heterogeneous payloads and time windows on efficiency and cost, providing feasible strategies for practical operations. Additionally, Yuan et al. [13] extended this line of inquiry by employing heterogeneous UAVs and riders for emergency medical delivery and developed a real-time joint scheduling method based on multi-agent deep reinforcement learning.

Furthermore, the application of drone technology in emergency medicine and healthcare logistics has garnered significant attention, with recent studies highlighting its potential to revolutionize last-mile delivery and emergency response systems. Johnson et al. [14] highlighted the expanding functions of drones in emergency medicine, noting that while current applications show promise, understanding end-user perceptions and acceptance remains a critical challenge for future integration. Liu et al. [15] demonstrated the feasibility of using drones to deliver Automated External Defibrillators (AEDs), presenting this strategy as a novel means to improve outcomes in out-of-hospital cardiac arrests (OHCA). Their review underscores the advantages of drone-based AED delivery over traditional emergency medical services (EMS), suggesting that drones can significantly enhance response times and patient prognosis.

Silvestri et al. [16] conducted a comprehensive scoping review categorizing drone applications in healthcare, including biomedical transport, AED delivery, logistics support, and air ambulance services. Their findings suggest that drones are increasingly embedded in routine healthcare operations, despite persistent challenges such as regulatory barriers and operational constraints. Zègre-Hemsey et al. [17] examined the barriers to real-time drone integration in emergency cardiac care across the United States, Sweden, and Canada, revealing that while notable progress has been made, regulatory complexity, safety concerns, and logistical coordination continue to hinder large-scale adoption.

Technological innovations aimed at enhancing drone performance for medical applications are also advancing. Hanafiah et al. [18] explored strategies for improving drone efficiency, emphasizing the importance of safety protocols, public awareness, and community engagement in facilitating broader deployment. Hemalatha et al. [19] introduced an AI-enabled MedDrone designed to improve healthcare logistics by facilitating autonomous, timely medication delivery, representing a significant step toward intelligent, automated medical supply chains.

In the context of emergency response, Kaduskar et al. [20] presented the SkyMedic, an autonomous ducted UAV engineered for rapid medical aid delivery. Its autonomous flight capabilities and integrated payload system are intended to reduce response times and en-

hance the effectiveness of critical care services. Complementing this work, Güzey et al. [21] investigated collaborative routing strategies involving UAVs and ground vehicles (UGVs) for disaster logistics, demonstrating how coordinated efforts can optimize delivery times during emergencies.

The broader research landscape reveals emerging thematic directions. Campbell et al. [22] conducted a study focused on drone healthcare applications: time-critical and non-time-critical deliveries, pointing to increasing diversification in the field. Bhattacharya et al. [23] further contributed a systematic review of drone customization and its implications for public health innovation, outlining future research directions to bridge existing gaps.

Despite these advancements, most existing studies are based on static or deterministic assumptions, providing insufficient attention to the stochastic disturbances prevalent in actual operations. Particularly in UAV delivery, low-altitude meteorological factors such as wind speed and direction significantly influence flight speed, energy consumption, and reachable range. Ignoring these uncertainties during the decision-making phase often leads to energy depletion or task failure during execution. Di Puglia Pugliese et al. [24] proposed a decomposition algorithm based on Benders combinatorial cuts to handle uncertain energy constraints influenced by atmospheric conditions. Lu et al. [25] constructed a path optimization model incorporating both static and dynamic wind fields for food delivery under extreme weather. Furthermore, Wu et al. [26] proposed a UAV energy model that accounts for dynamic flight speed and payload, reducing system energy consumption through multi-objective optimization. The authors [27] presented an improved Genetic Algorithm for path planning to reduce the UAV flight distance, which introduces the Teaching–Learning-based Optimization (TLBO) and local search optimization algorithms to improve convergence speed and the path solution. Although recent studies have explored wind-affected drone routing using scenario-based optimization and intermodal network designs under carbon constraints [28,29], a research gap remains in coupling rider–UAV collaborative dispatching with aerodynamic-based nonlinear power in a stochastic environment.

Overall, these studies collectively illustrate that drones are increasingly being integrated into last-mile delivery and emergency medical response, offering promising solutions to improve response times, enhance healthcare logistics, and address critical healthcare needs. However, challenges related to regulatory acceptance, operational safety, and technological optimization remain areas for ongoing research and development.

A synthesis of current research reveals that, despite progress in collaborative delivery and energy-aware modeling, systematic studies investigations integrating UAV–rider collaboration, nonlinear energy consumption and stochastic wind fields in urban instant delivery remain limited. In practice, riders and low-altitude UAVs are subject to fundamentally different sources of uncertainty: riders are primarily affected by road network structures and traffic flow, with their speed and energy being relatively insensitive to weather; conversely, a UAV's ground speed, endurance, and feasibility are highly dependent on stochastic wind factors. Neglecting these differences within a unified framework can lead to high risks of energy depletion for UAV paths during execution.

The main contributions of this paper are summarized as follows:

- (1) A nonlinear UAV energy consumption model is developed, incorporating payload mass, flight speed, and stochastic wind vectors, and integrated into a UAV–rider collaborative dispatching framework for urban emergency medicine delivery.
- (2) Within a two-stage stochastic programming formulation, the precise calculation of nonlinear energy consumption is embedded into the pricing sub-problem of the branch-and-price algorithm, avoiding the physical inaccuracies introduced by conventional linearization methods;

- (3) Through multi-scale comparative experiments, the impact of nonlinear energy modeling on delivery feasibility and scheduling robustness is quantitatively assessed, providing decision support for the safe and efficient operation of urban instant delivery systems.

3. Problem Description and Modeling

3.1. Problem Description

This paper investigates the UAV–rider collaborative dispatching problem (URCDP) considering energy consumption under stochastic wind conditions. The system consists of a distribution center, a fleet of ground riders, a fleet of UAVs, and a set of customers to be served. Given the highly variable wind fields characteristic of urban low-altitude environments, this study employs a two-stage stochastic programming approach for modeling. Ground riders are primarily influenced by road network traffic, while the interference from wind conditions is negligible. Therefore, the rider’s routing is treated as a first-stage deterministic decision, where wind effects are ignored. In contrast, UAVs flying at low altitudes experience ground speed, energy consumption, and maximum service radii that are significantly affected by wind speed and direction. Consequently, the UAV routes are modeled as second-stage scenario-dependent decisions, where flight plans are dynamically adjusted based on observed wind scenarios to adapt to environmental changes.

To ensure model tractability while preserving practical relevance, the model was formulated based on the following core assumptions:

- (1) Each customer is served exactly once by either a rider or a UAV. This binary assignment constraint is fundamental to UAV–ground collaborative routing models, ensuring clear demand fulfillment and simplifying task allocation logic [30].
- (2) Riders and UAVs depart from and return to the distribution center, without physical handover parcel handovers during routes. This assumption simplifies coordination complexity, a standard simplification in contemporary hybrid delivery systems where in-route transfers are logistically prohibitive [31].
- (3) UAVs start with a full battery charge. Loading/unloading times are neglected, while battery swapping time is treated as a fixed delay. This approach balances realism and computational efficiency, widely adopted in recent UAV energy-aware routing studies [32].
- (4) UAV energy consumption is a nonlinear function coupling payload, airspeed, and wind vectors. Recent studies confirm that ignoring wind and payload dynamics leads to unrealistic range predictions; thus, a nonlinear model is essential for accurate feasibility analysis [33].
- (5) Under all considered wind scenarios, UAV routes must satisfy the non-negative battery constraint. This robust constraint is critical for stochastic UAV routing to prevent energy depletion in uncertain environments [34].
- (6) Riders travel at a preset average speed, with wind effects deemed negligible. Traffic conditions dominate ground travel time variability, and this assumption aligns with state-of-the-art urban logistics models [35].
- (7) Early arrivals are permitted, while tardiness incurs a linear penalty cost. This setting models real-world service flexibility and is a standard practice in vehicle routing optimization for on-demand services [36].

Figure 1 illustrates the schematic of the delivery system. The system consists of four core components: a distribution center, ground riders, low-altitude UAVs, and customers. UAVs handle urgent, long-distance deliveries, while ground riders serve areas with flight restrictions or non-urgent orders. Both fleets operate independently from the same depot. The system follows a two-stage stochastic programming framework: riders execute

deterministic path plans (first-stage decision) with speed and energy insensitive to wind, while UAVs dynamically adjust flight paths based on specific wind scenarios (second-stage recourse decision) to ensure delivery feasibility under nonlinear energy constraints.

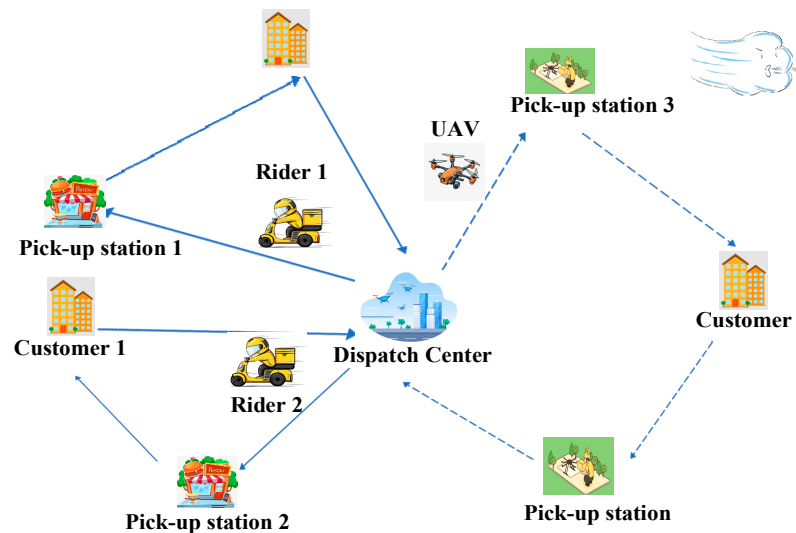


Figure 1. Schematic diagram of the UAV–ground rider collaborative dispatch system.

3.2. Model Formulation

3.2.1. Description of Model Symbols

To facilitate the model description, the relevant parameters, and variables are defined as shown in Table 1.

Table 1. Symbol definitions.

Symbol	Description	Unit
N	Set of nodes, $N = \{0, 1, \dots, n\}$, where 0 denotes the depot	—
D	Set of customer nodes, $D = N \setminus \{0\}$	—
U	Set of available UAVs, $u \in U$	—
R	Set of available delivery riders, $r \in R$	—
S	Set of stochastic wind scenarios, $s \in S$	—
A	Set of feasible arcs between nodes, $A = \{(i, j) \mid i, j \in N, i \neq j\}$	—
Ω	Set of all feasible delivery routes	—
H	Set of delivery resources, $H = U \cup R$	—
$ng(i)$	Set of neighboring nodes of node i	—
S_{sp}	Set of customer subsets for subset row inequalities	—
Parameter	Description	Unit
d_{ij}	Euclidean distance from node i to j	km
$[e_i, l_i]$	Service time window for customer i	min
s_i	Service duration for customer i	min
q_i	Demand weight of customer i	kg
t_{ij}^r	Travel time of rider r from node i to j	min
Q_{max}^r	Maximum load capacity of rider r	kg
v_r	Average travel speed of rider r	km/h
c_r	Unit distance cost of rider r	RMB/km
c_f^r	Unit time fixed cost of rider r	RMB/h
$t_{ij}^{u,s}$	Flight time of UAV u from node i to j under wind scenario s	min
Q_{max}^u	Maximum load capacity of UAV u	kg
v_{air}	Cruise airspeed of UAV (no-wind condition)	km/h
v_w^s	Environmental wind speed under scenario s	m/s
θ_s	Wind direction angle under scenario s	rad

Table 1. Cont.

Parameter	Description	Unit
α_{ij}	Heading angle of UAV flying along arc (i, j)	rad
φ_{ij}^s	Relative angle between wind direction and UAV heading, $\varphi_{ij}^s = \theta_s - \alpha_{ij}$	rad
$v_g^{ij,s}$	Actual ground speed of UAV along arc (i, j) under scenario s	km/h
m_t	Empty weight of UAV	kg
m	Total weight of UAV, $m = m_t + q$, (where q is the current payload)	kg
κ	Rotor induced power coefficient	—
T	Approximate thrust of UAV	N
ω	Angular velocity of UAV rotor	rad/s
c_1	UAV profile power coefficient	—
v_{tip}	Tip speed of UAV rotor blade	m/s
c_2	UAV gravity-related power coefficient	—
P_{int}	Baseline power consumption of UAV on-board equipment	W
c_u	Unit energy cost of UAV	RMB/kWh
c_f^u	Fixed cost per mission for UAV u	RMB
c_c	Battery replacement cost per time for UAV	RMB/time
δ	Empty weight equivalent load factor of UAV	—
k	Energy consumption proportional coefficient of UAV	—
p_s	Probability of scenario s occurring, $\sum_{s \in S} p_s = 1$	—
α	Unit overtime penalty cost	RMB/min
β	Unit waiting penalty cost	RMB/min
M	A sufficiently large constant	—
m_h	Upper limit of available quantity of resource h	—
Binary	Description	Unit
$x_{ij}^{u,s}$	Binary variable : 1 if UAV u traverses arc (i, j) under scenario s ; 0 otherwise	—
y_{ij}^r	Binary variable : 1 if rider r traverses arc (i, j) ; 0 otherwise	—
z_i^u	Binary variable : 1 if customer i is served by UAV; 0 otherwise	—
z_i^r	Binary variable : 1 if customer i is served by a rider, 0 otherwise	—
$T_i^{u,s}$	Arrival time of UAV u upon arrival at node i under scenario s .	min
T_i^r	Arrival time of rider r at node i	min
$Q_i^{u,s}$	Remaining payload of UAV u when leaving node i under scenario s	kg
$Re_i^{u,s}$	Overtime duration at customer i under scenario s	min
L_i^s	Overtime duration at customer i under scenario s	min
W_i^s	Waiting duration of UAV at customer i under scenario s	—
y^u	Binary variable : 1 if UAV u is activated; 0 otherwise	h
T^r	Total working duration of rider r	—
S_u^s	Binary variable : 1 if UAV u replaces battery under scenario s ; 0 otherwise	—
θ_p^h	Binary variable : 1 if resource h selects route p ; 0 otherwise	—
α_p^i	Binary parameter : 1 if route p serves customer i ; 0 otherwise	—
c_p^h	Total cost for resource h to execute route p	RMB
λ_i	Dual variable corresponding to the customer coverage constraint	—
π_h	Dual variable corresponding to the resource quantity constraint	—
φ_S	Dual variable corresponding to the subset row inequality cut constraint	—
$\bar{c}_{ph}$	Reduced cost for resource h to execute route p	RMB
$\bar{c}_i$	Cumulative reduced cost of the path from depot to node i	RMB
τ_i	Arrival time at node i in the labeling algorithm	min
ϵ_i	Remaining battery energy upon arriving at node i in the labeling algorithm	wh

3.2.2. Stochastic Wind Scenario Generation and Ground Speed Calculation

Wind conditions are represented as scenario-based stochastic variables that influence UAV flight performance. In particular, wind uncertainty affects (i) the effective ground speed of UAVs, which alters travel time between service nodes; (ii) the energy consumption rate, due to additional thrust required to counteract headwinds and stabilize flight under crosswinds; and (iii) the maximum service radius, as increased energy consumption reduces the feasible flight range. These stochastic effects are incorporated into the second-

stage recourse decisions to ensure robust and energy-feasible UAV routing under varying wind scenarios.

Urban low-altitude wind fields are characterized by high spatiotemporal stochasticity. In this study, the wind speed v_w is modeled using a Weibull distribution, while the wind direction angle θ_s is assumed to follow a uniform distribution over $[0, 2\pi)$. To strike a balance between computational efficiency and the accurate representation of stochastic features, a scenario reduction technique based on the Kantorovich distance is employed. Initially, 1000 wind condition samples are generated via Monte Carlo simulation, encompassing a wide range of wind speed and direction combinations. A backward reduction method is applied to minimize the probabilistic distance, selecting $|S|=5$ most representative typical scenarios. Each scenario $s \in S$ includes a deterministic wind speed v_w^s , wind direction θ_s and its occurrence probability p_s , satisfying $\sum_{s \in S} p_s = 1$. This methodology preserves the statistical properties of the original distribution while maintaining the stochastic programming model at a tractable scale. To determine an appropriate number of scenarios, preliminary experiments were conducted to assess the sensitivity of the objective function to varying scenario counts. The results indicate that when the number of typical scenarios $|S| \geq 5$, the objective value tends to stabilize, while the computation time increases significantly with more scenarios. Considering the trade-off between solution quality and computational efficiency, this paper sets $|S|=5$ as the number of stochastic wind scenarios. Synthetic wind speeds are generated using a Weibull distribution with shape parameter $k = 2.0$ and scale parameter $c = 6.5$ m/s, representing typical urban low-altitude wind conditions. To incorporate wind uncertainty into the stochastic programming model, the continuous Weibull wind speed distribution is discretized into five representative scenarios using an equal-probability (quantile-based) method. The cumulative distribution function is divided into five intervals with equal probability (0.2), and the midpoint wind speed of each interval is selected as the representative value.

This discretization preserves the key characteristics of the original distribution, including probability mass, mean, and variability, while ensuring that both low- and high-wind conditions are adequately represented.

It should be noted that the wind scenarios considered in this study are constrained within the UAV's normal operational range. Extreme or rare weather events exceeding this threshold are intentionally excluded. This is because such conditions would fundamentally compromise the structural safety of the UAV, triggering a mandatory 'grounding' or suspension of operations—a 'go/no-go' tactical decision that falls outside the scope of the routing optimization problem presented here. By focusing on the operable range, the model provides robust and applicable scheduling for the vast majority of delivery windows.

The actual flight time of a UAV depends on its speed relative to the ground $v_g^{ij,s}$. This paper utilizes the wind triangle geometric model for calculation. Let the UAV's cruising airspeed in calm air be a constant v_u , in a wind scenario s , when the UAV flies along an arc (i, j) with an azimuth angle of α_{ij} , the angle between the wind direction and the flight heading is $\phi_{ij}^s = \theta_s - \alpha_{ij}$. According to the law of vector addition, the UAV's actual ground speed $v_g^{ij,s}$ is given by [28]:

$$v_g^{ij,s} = \sqrt{v_{air}^2 - (v_w^s \sin \phi_{ij}^s)^2} + v_w^s \cos \phi_{ij}^s \quad (1)$$

This formula accurately characterizes the loss of ground speed components due to crosswinds. The acceleration and deceleration phases are neglected as they account for a

minor fraction of the total delivery time at cruise altitude. Consequently, the flight time for a UAV to traverse arc (i, j) under scenario s is:

$$t_{ij}^{u,s} = \frac{d_{ij}}{v_{ij}^{u,s}} \quad (2)$$

Based on aerodynamic principles, a nonlinear energy consumption function is constructed by coupling payload, speed, and wind conditions [30]. The instantaneous power P_u of the UAV is a nonlinear function of the total weight m and the air speed v_{air} . This paper adopts a modified integrated power model:

$$P_u(m, v_u) = \kappa \cdot T \cdot \omega + c_1 (v_{tip}^3 + 2v_u^2 v_{tip}) + c_2 \cdot m \cdot g + P_{int} \quad (3)$$

m is the sum of the empty weight m_t and the current payload q_l of UAV; T represents the thrust approximation; P_{int} denotes the base power consumption of onboard equipment. We explicitly assume that the UAV maintains a constant cruising airspeed (v_{air}). This assumption simplifies the power model while effectively capturing how different wind realizations alter the ground speed vector and the corresponding energy consumption rate, which is the primary driver of mission failure risk. Thus, in scenario s , the total energy consumption for a UAV carrying payload q across arc (i, j) is:

$$E_{i,j}^{u,s}(q) = P_u(m_t + q, v_u) \times t_{i,j}^{u,s} \quad (4)$$

where energy consumption is measured in Joules (J).

It should be noted that the UAV power model employed in this study is applicable to typical multi-rotor UAVs, with a structure based on the integrated power decomposition forms commonly used in existing literature. The parameters κ , c_1 , c_2 and P_{int} are used to characterize rotor-induced power, profile power, gravity-related power, and the base power consumption of onboard equipment, respectively. Their orders of magnitude and value ranges refer to the experimental calibration results in the research of Dodge et al. [29] and Wu et al. [26]. In the numerical experiments of this paper, these parameters remain constant under unified calibration conditions to reflect the impact of different wind conditions and payload variations on relative energy levels.

3.2.3. Two-Stage Stochastic Programming Model

In this framework, the first-stage decisions focus on the stability of the rider's skeleton route, while the second-stage actions allow for adaptive UAV adjustments. The model's first stage is adjusted by the expected performance of the second stage, ensuring that the final schedule is feasible across the entire distribution of wind scenarios. Within the two-stage stochastic programming framework, decision variables related to riders (y_{ij}^r , z_i^r , T^r) are defined as first-stage decisions. Their values remain consistent across all wind scenarios, thereby implicitly satisfying non-anticipation constraints. Conversely, variables related to UAV paths and energy consumption ($x_{ij}^{u,s}$, z_i^u , $T_i^{u,s}$, $Q_i^{u,s}$, $Re_i^{u,s}$, L_i^s , W_i^s , S_u^s) are categorized as second-stage decisions, allowing for dynamic adjustments based on specific wind scenarios. The objective of this model is to minimize the sum of deterministic rider costs and the expected operational costs of the UAVs, as shown in Equation (5). The mixed-integer programming model [M] is established as follows:

$$\begin{aligned} \min C = & \sum_{r \in R} \sum_{(i,j) \in A} c_r d_{ij} y_{ij}^r + \sum_{u \in U} c_f^u y^u + \sum_{r \in R} c_f^r T^r + \\ & \sum_{s \in S} p_s \left[\sum_{u \in U} \sum_{(i,j) \in A} c_u E_{i,j}^{u,s} x_{ij}^{u,s} + c_c \sum_{u \in U} S_u^s \right] + \alpha \sum_{i \in N} L_i^s + \beta \sum_{i \in N} W_i^s \end{aligned} \quad (5)$$

$$s.t. \sum_{u \in U} Z_i^u + \sum_{r \in R} z_i^r = 1, \forall i \in D \quad (6)$$

$$\sum_{i \in N} x_{ij}^{u,s} = Z_i^u, \forall i \in D, u \in U, s \in S \quad (7)$$

$$\sum_{j \in N} y_{ij}^r = Z_i^r, \forall i \in D, r \in R \quad (8)$$

$$\sum_{j \in N} x_{ij}^{u,s} = \sum_{j \in N} x_{ji}^{u,s}, \forall i \in N, u \in U, s \in S \quad (9)$$

$$\sum_{j \in N} y_{ij}^r = \sum_{j \in N} y_{ji}^r, \forall i \in N, r \in R \quad (10)$$

$$\sum_{j \in N} x_{0j}^{u,s} \leq 1, \sum_{j \in N} x_{j0}^{u,s} \leq 1, \forall u \in U, s \in S \quad (11)$$

$$\sum_{j \in N} y_{0j}^r \leq 1, \sum_{j \in N} y_{j0}^r \leq 1, \forall r \in R \quad (12)$$

$$T_j^{u,s} \geq T_i^{u,s} + t_{ij}^{u,s} - M(1 - x_{ij}^{u,s}), \forall (i, j) \in A, u \in U, s \in S \quad (13)$$

$$T_j^r \geq T_i^r + t_{ij}^r - M(1 - y_{ij}^r), \forall (i, j) \in A, r \in R \quad (14)$$

$$T_i^{u,s} \geq e_i z_i^u, T_i^{u,s} \leq l_i + M(1 - z_i^u), \forall i \in D, u \in U, s \in S \quad (15)$$

$$T_i^r \geq e_i Z_i^r, T_i^r \leq l_i + M(1 - Z_i^r), \forall i \in D, r \in R \quad (16)$$

$$L_i \geq T_i - l_i, L_i \geq 0, \forall i \in D \quad (17)$$

$$W_i \geq e_i - T_i, W_i \geq 0, \forall i \in D \quad (18)$$

$$Q_0^{u,s} = \sum_{i \in D} q_i z_i^u, \forall u \in U, s \in S \quad (19)$$

$$Q_j^{u,s} \leq Q_i^{u,s} - q_j x_{ij}^{u,s} + M(1 - x_{ij}^{u,s}), \forall (i, j) \in A, u \in U, s \in S \quad (20)$$

$$0 \leq Q_i^{u,s} \leq Q_{max}^u, \forall i \in N, u \in U, s \in S \quad (21)$$

$$Re_j^{u,s} \leq Re_i^{u,s} - E_{ij}^{u,s} + M(1 - x_{ij}^{u,s}), \forall i \in N, u \in U, s \in S \quad (22)$$

$$0 \leq Re_i^{u,s} \leq E_{max}^u, \forall i \in N, u \in U, s \in S \quad (23)$$

$$x_{i,j}^{u,s}, y_{i,j}^r, z_i^u, z_i^r, y^u, S_u^s \in \{0, 1\}, \quad Re_i^{u,s}, T_i^{u,s}, T^r, T_i^r, Q_i^{u,s}, L_i^s, W_i^s \geq 0 \quad (24)$$

Equation (6) ensures each customer is served exactly once by either a UAV or a rider; Equations (7) and (8) maintain consistency between customer service status and path decision variables (i.e., linking whether a customer is served to the actual routing choices); Equations (9) and (10) are flow balance constraints, guaranteeing path continuity as resources (UAVs/riders) must leave any node they enter; Equations (11) and (12) restrict each rider or UAV to executing at most one delivery route per mission; Equations (13) and (14) govern time accumulation along paths, and adopting the Big-M method to connect binary decision variables with arrival times at nodes; Equations (15) and (16) define time window constraints, ensuring arrival at customer nodes fall within specified intervals while accounting for service time and potential waiting time (if arriving early); Equations (17) and (18) calculate timeout durations (for late arrivals) and waiting durations (for early arrivals), with timeout penalties introducing differentiated priorities for customers; Equations (19)–(21) enforce UAV payload limits, total weight of parcels carried cannot exceed the UAV's maximum load capacity; Equations (22) and (23) monitor energy consumption and remaining battery levels, ensuring the total energy used by a UAV does not exceed its full battery capacity at any point during flight; and Equation (24) defines the value domains for all decision variables.

3.3. Model Reformulation Based on Dantzig–Wolfe Decomposition

The URCDP model is a typical path-based combinatorial optimization problem. When solved directly via mixed-integer programming (MIP), its computational complexity increases exponentially with the growth of the customer scale. To enhance both solvability and computational efficiency, the Dantzig–Wolfe decomposition algorithm is adopted. This approach decomposes the original problem into a master problem (MP) and a resource-constrained shortest path pricing sub-problem, which effectively reduces the problem dimensionality while maintaining solution quality.

3.3.1. Set Covering Master Problem Formulation

The core objective of the master problem is to achieve full coverage of customer demands through a combination of paths while minimizing total operational costs T .

The set covering master problem (MP) is formulated as follows:

$$\min \sum_{h \in H} \sum_{p \in \Omega} c_p^h \theta_p^h \quad (25)$$

$$\sum_{h \in H} \sum_{p \in \Omega} \alpha_p^i \theta_p^h = 1, \forall i \in D \quad (26)$$

$$\sum_{p \in \Omega} \theta_p^h \leq m_h, \forall h \in H \quad (27)$$

$$\theta_p^h \in \{0, 1\}, h \in H, p \in \Omega \quad (28)$$

Equation (26) ensures that each customer's demand is served exactly once by exactly one delivery resource. Equation (27) represents the availability limit for each type of delivery resource, where m_h denotes the upper bound on the available quantity of resource h .

Since the set of feasible paths Ω remains extremely large, directly solving the master problem is computationally prohibitive. Therefore, based on the column generation algorithm, the variable θ_p^h is linearly relaxed, and only a subset of paths $\Omega' \subseteq \Omega$ is considered, resulting in the restricted master problem (RMP).

3.3.2. Subset–Row Inequalities

To enhance the tightness of the linear relaxation for the restricted master problem (RMP), subset–row inequalities (SRIs) based on Chvátal–Gomory cuts are introduced [37]. These are dynamically embedded into the master problem iteration process as valid cutting planes. For any customer subset $\hat{S} \subseteq D$, let m be the maximum number of tasks covered by a single path, defined as $m = \max_{p \in \Omega} |\{i \in D | \alpha_p^i = 1\}|$. The subset–row inequality is expressed as follows:

$$\sum_{p \in \Omega'} \frac{1}{m} \sum_{i \in \hat{S}} \alpha_p^i \theta_p^h \leq \frac{|\hat{S}|}{m}, \forall h \in H \quad (29)$$

In this inequality, θ_p represents the linear relaxation value (selection degree) of the path variable, and $\hat{S}$ denotes the task subset. The left side of the inequality quantifies the coverage contribution of the current path subset to the customer subset $\hat{S}$, while the right side serves as the lower bound for the minimum number of paths required to cover $\hat{S}$. If this inequality is violated during the iteration process, it indicates insufficient coverage capability of the current path combination; thus, it must be added to the RMP as a cutting plane until no further inequalities are violated, where $|\dots|$ denotes the cardinality (number of elements) of the set. Thus, the model remains differentiable in its discrete domain.

3.3.3. Pricing Sub-Problem Modeling

The core of the pricing sub-problem is to identify feasible paths with a negative reduced cost guided by the dual variables of the master problem to expand the master problem's path subset Ω' . Let λ_i be the dual variable corresponding to the customer coverage constraint (26), π_h be the dual variable for the resource quantity constraint (27), $\hat{S} \in S_{sp}$ be the set of customer subsets for the subset–row inequalities, and $\varphi_{\hat{S}}$ be the dual variable for the corresponding cutting plane constraint (29). The reduced cost for using path p with resource h can be expressed as in Equation (30):

$$\bar{c}_{ph} = c_p^h - \sum_{i \in D} \lambda_i \alpha_p^i - \pi_h - \sum_{\hat{S} \in S_{sp}} \varepsilon(\hat{S}, p) \varphi_{\hat{S}} \quad (30)$$

where $\varepsilon(\hat{S}, p) = 1$ if path p covers the task subset $\hat{S}$, and 0 otherwise. The pricing sub-problem can be transformed into a shortest path problem with resource constraints (SPPRC). The objective function is $\min \bar{c}_{ph}$, and the constraints include the resource limits defined in Section 3.2.3, such as time windows, payload capacity, and battery energy. If the pricing sub-problem yields a path with a negative reduced cost, it is added to the master problem's path subset for a new iteration; if no such path exists, the current master problem solution is the optimal solution for the original problem.

4. Improved Branch-and-Price Algorithm

Given that the UAV–rider collaborative dispatching problem (URCDP) involves non-linear energy dynamics, multiple constraints, and large-scale integer programming, commercial solvers often fail to obtain exact solutions within a reasonable time. To address this, we propose an improved branch-and-price (IBP) algorithm integrating hybrid pricing and ng-path relaxation, achieving over 40% higher efficiency on large-scale instances than conventional exact solvers and standard branch-and-price methods. The algorithm transforms nonlinear energy constraints into dynamic evaluations within the pricing sub-problem, eliminating linearization errors while preserving physical accuracy. Key enhancements include a multi-dimensional label dominance rule to reduce state–space complexity, a hybrid heuristic–exact pricing strategy to balance speed and accuracy, an arc–resource coupling branching rule to control tree growth, and a wind-aware greedy heuristic to accelerate column generation convergence, ensuring both computational efficiency and modeling fidelity.

4.1. Algorithm Overview and Process

The algorithm adopts a framework that integrates column generation with branch-and-bound as Figure 2 shows. The master problem is tasked with selecting the optimal combination of paths from the generated set to satisfy customer coverage and flow balance constraints. After relaxing the integer constraints, a linear programming master problem is derived, whose dual solutions are then used to guide the sub-problem in generating new paths. The pricing sub-problem aims to find new path columns with a negative reduced cost. For the two-stage stochastic programming model proposed in this study, the sub-problem is decomposed into a rider sub-problem and a UAV sub-problem. The core steps are as follows:

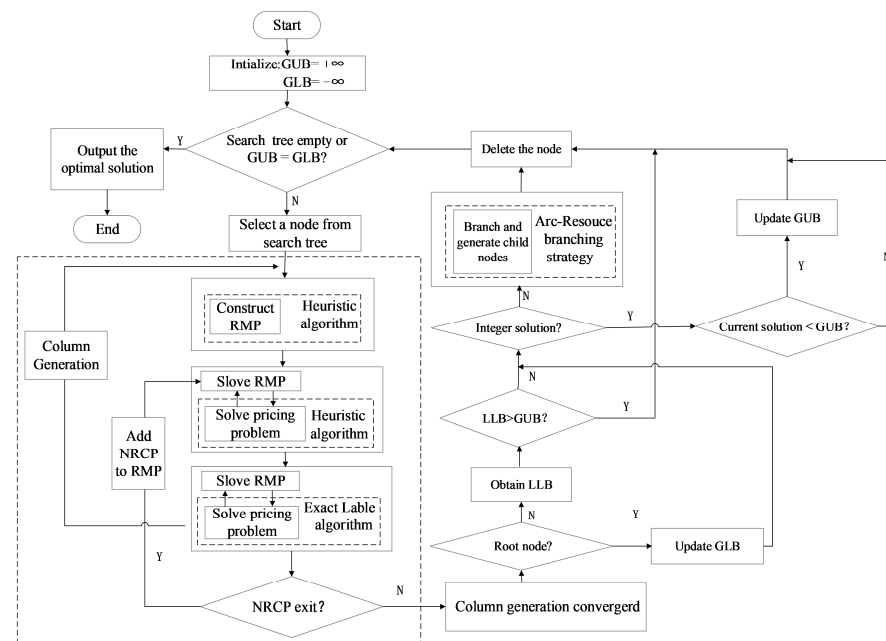


Figure 2. Improved branch-and-price framework.

Step 1: Rapidly construct a set of feasible paths that satisfy basic resource and time window constraints using a comprehensive heuristic that accounts for distance, congestion, and wind conditions. These paths constitute the initial column set for the restricted master problem (RMP), ensuring all order demands are covered.

Step 2: Solve the RMP and its dual problem based on the current path set, and pass the resulting dual variables to the pricing sub-problem.

Step 3: Employ a hybrid pricing sub-problem solving mechanism: First, a heuristic search is performed to quickly identify paths with negative reduced cost. If no such paths are found, an exact labeling algorithm is invoked to conduct an exhaustive search, thereby guaranteeing the optimality of the RMP relaxation or convergence.

Step 4: If new paths with a negative reduced cost exist are identified, add them to the RMP and return to Step 2. Otherwise, the column generation has converged, yielding the optimal linear relaxation solution for the current node.

Step 5: If the current solution contains fractional variables, apply a branching strategy based on arc–resource coupling to generate child nodes, and repeat Steps 2–4 for each child node.

Step 6: When all search nodes either satisfy integrality or are proven infeasible, the algorithm terminates and outputs the global optimal path combination.

4.2. Initial Solution Generation Strategy

The quality of the initial solution determines the initial lower bound of the restricted master problem (RMP) and the convergence speed of column generation. In this paper, a greedy insertion heuristic based on task priority is designed to generate high-quality initial paths. The specific steps are as follows:

Step 1: Starting from the distribution center, compute a priority value for each unserved customer by integrating remaining time, unit distance cost, and wind impact.

Step 2: Select the node from the candidate set that yields the smallest increase in unit distance cost and attempt to insert it into the current path.

Step 3: Determine if the path satisfies time window, load, battery, and wind constraints after insertion; if not, adjust the position or reselect a node.

Step 4: When current path resources are exhausted or all nodes have been visited, terminate the current path and generate a new one.

Step 5: Consolidate the generated UAV and rider paths into a complete dispatching plan, which serves as the initial column set for column generation.

We used a construction heuristic strategy based on local feasible insertion. By implementing candidate set restrictions and constraint feasibility checks, it ensures the quality of initial paths while shortening subsequent iteration times.

4.3. Exact Solving Strategy for Nonlinear Energy Consumption in the Pricing Sub-Problem

The pricing sub-problem is modeled as a shortest path problem with resource constraints (SPPRC). Unlike existing studies that typically use McCormick envelopes or piecewise linearization to handle nonlinear energy constraints, this paper utilizes the forward expansion characteristics of the label-setting algorithm on a topological sort to transform complex nonlinear function calculations into deterministic numerical evaluations during state transitions.

4.3.1. Label Definition and State Space

A label L_i is defined as the state vector associated with node i of a partial path from the distribution center 0 to node i . It represents the state of a partial path in the labeling algorithm. $L_i = (\eta_i, \tau_i, q_i, \epsilon_i, \bar{c}_i, v_i)$. η_i : the preceding node of node i ; τ_i : arrival time at node i ; q_i : current load when leaving node i ; ϵ_i : remaining battery energy upon arrival at node i ; $\bar{c}_i$: cumulative reduced cost of the current path; v_i : set of already visited nodes.

4.3.2. Nonlinear Exact Calculation Based on State Transitions

The core of the labeling algorithm lies in the fact that when a state is expanded from node i to j , all state variables (τ_i, q_i) are known scalars for the current local decision. Consequently, nonlinear variables that are highly coupled in mixed-integer programming (MIP) degrade into constant parameters at the moment of label extension. Specifically, the resource update and feasibility verification logic are as follows:

$$q_j = q_i - d_j \quad (31)$$

$$\Delta E_{ij}^s = P_u \left(m_t + q_i, v_g^{ijs} \right) \cdot t_{ij}^{u,s} \quad (32)$$

$$\epsilon_j = \epsilon_i - \Delta E_{ij}^s \quad (33)$$

$$\bar{c}_j = \bar{c}_i + \left(c_u \cdot \Delta E_{ij}^s - \pi_i \right) \quad (34)$$

In these equations, q_j represents the delivery demand at node j , and if $q_j < 0$, the expansion is infeasible. Equation (32) provides a precise evaluation of nonlinear energy consumption, where $P_u(\cdot)$ is the nonlinear power function. Since input variables are deterministic values at the time of expansion, ΔE_{ij}^s can be calculated exactly without auxiliary variables or linear inequalities. Equation (33) determines battery state updates; if $\epsilon_j < 0$, the path is pruned. Here, π_i is the dual variable passed from the master problem, and c_u is the unit energy cost.

4.3.3. Label Dominance Rules

To avoid state redundancy and invalid expansion, the following dominance rule is applied: assume two labels, $L_1(j, t_1, e_1, q_1, \bar{c}_1, V_1)$ and $L_2(j, t_2, e_2, q_2, \bar{c}_2, V_2)$, reach the same node j ; L_1 dominates L_2 if the following conditions are met:

$$(1) \tau_1 \leq \tau_2; (2) \epsilon_1 \geq \epsilon_2; (3) \bar{c}_1 \leq \bar{c}_2; (4) V_1 \supseteq V_2.$$

Note: Although the energy function is nonlinear, within the physical parameters set in this study, energy consumption E is monotonically non-decreasing with flight time t and weight m . Since the distance $d_{ij} > 0$, path expansion necessarily leads to an increase in cumulative energy consumption and cost. Thus, the multi-dimensional resource dominance rules still satisfy the monotonicity assumption under nonlinear conditions, ensuring that pruning does not omit the global optimal solution.

4.3.4. Ng-Path Relaxation Strategy

To control the growth of label quantities, an ng-path relaxation strategy is introduced in the pricing sub-problem. During expansion, strict access restrictions are applied only to certain key nodes, while limited repeat visits are allowed for nodes outside the neighborhood: (1) A neighborhood set $ng(i)$, is constructed for each customer, consisting of the γ nearest nodes. (2) Labels record only visited nodes within $ng(i)$, reducing state space complexity from $O(n \cdot 2^n)$ to $O(n \cdot 2^\tau)$ ($\tau \ll n$). (3) Non-repeatable access constraints are enforced only for nodes within $ng(i)$. (4) If the completed path contains duplicate customers, it is filtered during a post-processing stage. This strategy significantly reduces label volume without excluding the optimal solution.

4.3.5. Symmetry-Breaking Strategy

$x_{ij}^k x_{ij} = \sum_{k \in K} x_{ij}^k$. To mitigate symmetry caused by identical UAVs and courier riders in the fleet—which can generate redundant nodes in the branch-and-bound tree—two symmetry-breaking mechanisms are incorporated into the IBP algorithm. First, instead of branching on the binary decision variable for a specific vehicle k , branching is performed on the aggregated arc flow. This treats identical vehicles as a shared resource pool and avoids redundant branches that differ only by vehicle assignment. Second, a hierarchical ordering is imposed on identical vehicles, such that a vehicle with a higher index can be used only if all lower-indexed vehicles are assigned to non-empty routes. This rule ensures a unique representative solution among symmetric permutations.

4.4. Hybrid Sub-Problem Solving Strategy

In the branch-and-price framework, the solving efficiency of the pricing sub-problem is a critical factor affecting overall algorithm performance. Although exact label-based algorithms guarantee finding the path with the optimal reduced cost, their computational complexity rises sharply as the problem scale increases. Therefore, this paper adopts a hybrid strategy combining heuristic search with the exact labeling algorithm to enhance column generation efficiency while maintaining algorithmic accuracy. During the early stages of column generation, dual variables often oscillate significantly, making it unnecessary to find the global optimum in every iteration. This paper utilizes a “remove-and-insert” strategy based on local search, with the following operations:

(1) Random Removal Operator

Based on a preset removal ratio, q customer nodes are selected and removed from the current path solution sol_{cul} to form a partial path P_p and a set of nodes to be inserted N_u . This operation enhances search diversity by disrupting the existing path structure.

(2) Greedy Insertion Operator

For each node $i \in N_u$, the change in reduced cost $\Delta \bar{c}_{ph}$ is calculated for different insertion positions h , where $\Delta \bar{c}_{ph} = \bar{c}_{ph}^i - \bar{c}_{ph}$. Positions that minimize $\Delta \bar{c}_{ph}$ and satisfy time window, load, energy, and wind constraints are prioritized.

(3) Solution Acceptance Mechanism Based on the Metropolis Criterion

To balance search depth and breadth, a simulated annealing acceptance mechanism is introduced. Given an initial temperature T_0 and a cooling coefficient α , in the k th iteration, the acceptance probability P for a new solution sol_{new} is defined as in Equation (35):

$$p = \begin{cases} 1, & \text{if } obj(sol_{new}) < obj(sol_{cul}) \\ e^{\frac{-obj(sol_{new}) - obj(sol_{cul})}{T_k}}, & \text{otherwise} \end{cases} \quad (35)$$

where $obj(\cdot)$ represents the reduced cost objective value of the path and T_k is the temperature at the k th iteration. If the new solution is superior, it is accepted immediately; otherwise, it is accepted with probability P . The temperature is updated via $T_{k+1} = \alpha \cdot T_k$ after a certain number of iterations.

(4) Heuristic algorithm pseudo-code

Algorithm 1 shows the pseudo-code of heuristic search algorithm for the pricing sub-problem.

Algorithm 1: Heuristic Search Algorithm for the Pricing Sub-problem

Input: Initial feasible solution sol , search scale $scal$, initial temperature T_0 , cooling rate α , maximum iterations $\max_{iter}$.

Output: Optimal solution sol_{best}

```

1:  $sol_{best} \leftarrow sol$ 
2:  $curr_{sol} \leftarrow sol$ 
3:  $T \leftarrow T_0$ 
4:  $iter \leftarrow 0$ 
5: while  $iter < \max_{iter}$  do
6:    $curr_{sol}' \leftarrow curr_{sol}$ 
7:   Randomly remove  $scal$  nodes from  $curr_{sol}'$  to obtain node set  $N_u$ .
8:   For each node  $i \in N_u$ , calculate  $\Delta \bar{c}_{ph}$  for all feasible positions.
9:   Sort by  $\Delta \bar{c}_{ph}$  and insert node  $i$  into the optimal position.
10:  if  $obj(sol_{new}) > obj(curr_{sol}')$  then
11:     $curr_{sol} \leftarrow curr_{sol}'$ 
12:    if  $obj(sol_{new}) < obj(curr_{sol}')$  then
13:       $sol_{best} \leftarrow curr_{sol}$ 
14:    end if
15:  else
16:    Calculate probability  $P$ , generate random number  $r \in [0, 1]$ 
17:    if  $r < P$  then
18:       $curr_{sol} \leftarrow curr_{sol}'$ 
19:    end if
20:  end if
21:   $T \leftarrow \alpha \cdot T$ 
22:   $iter \leftarrow iter + 1$ 
23: end while
24: return  $sol_{best}$ 

```

4.5. Branching Strategy

When the restricted master problem (RMP) yields a fractional solution that violates integrality constraints, branching is required. Given the heterogeneity of riders and UAVs, an arc-resource coupling branching strategy is employed:

Let $x_{ij}^h (h \in \{U, R\})$ represent the relaxed solution for resource h traversing arc (i, j) . Branching is performed on the fractional variable x_{ij}^h closest to 0.5. In one child node, $x_{ij}^h = 0$ is enforced, prohibiting resource h from traversing arc (i, j) , and removing all paths containing this arc. In the other child node, $x_{ij}^h = 1$ is enforced, requiring resource h to traverse arc (i, j) and retaining only paths containing this arc.

This strategy effectively avoids the excessively wide branching trees associated with traditional path-variable branching while preserving the structural characteristics of the sub-problem.

5. Numerical Experiments and Analysis

To evaluate the effectiveness and applicability of the proposed model and the improved branch-and-price (IBP) algorithm, numerical experiments are conducted from three perspectives: computational performance, collaborative advantages of the model, and key parameter sensitivity. To ensure the fairness and reliability of the comparison results, several rigorous control measures were implemented. All experiments are implemented on a macOS platform with an Apple M1 octa-core processor and 16 GB RAM. The algorithms are coded in Python 3.11, and the linear relaxation master problem is solved by Gurobi Optimizer 10.0. All benchmark methods—including the standard branch-and-price and heuristic approaches—are executed under the same hardware and software environment. In addition, UAV physical parameters, wind scenario distributions, and customer locations remain identical across all models. When comparing the two-stage stochastic model with its deterministic counterpart, the deterministic model uses the expected values of stochastic parameters, ensuring equivalent information conditions. Finally, a uniform optimality tolerance of 10^{-4} and a time limit of 3600 s are applied to all instances to guarantee directly comparable computational results.

5.1. Test Instance Design

Due to the absence of standardized test instances for the UAV–rider collaborative dispatching problem (URCDP) in the existing literature, this study constructs experimental instances by adapting the EVRPTW benchmark instances developed by Schneider et al. [38] and integrating real-world urban instant delivery data. The instances are categorized into three scales based on the number of customer nodes: small ($n = 25$), medium ($n = 50$), and large ($n = 100$). Based on node distribution characteristics, they are divided into three types: clustered (Type C), random (Type R), and mixed (Type RC). The naming convention follows the “Distribution Type + Node Count – Group Number” rule; for example, C25-1 represents the first group of instances with 25 customer nodes under a clustered distribution.

To evaluate the effectiveness of the IBP algorithm, three benchmark approaches are introduced for comparison: the traditional branch-and-price algorithm, the Gurobi commercial solver, and the adaptive large neighborhood search (ALNS) algorithm as comparison benchmarks. Given the significant influence of key parameters on algorithm performance, an orthogonal experimental method is used to complete parameter pre-tuning on small-scale instances. The average speed of riders is set to 25 km/h with a maximum payload of 15 kg. Five representative wind scenarios are generated based on a Weibull distribution, with wind speeds ranging from 0 to 12 m/s. The final determined core parameters are shown in Table 2.

Table 2. Key parameters.

Symbol	Definition	Value
c_f^r	Rider fixed cost per unit time	30 RMB/h
c_f^u	UAV fixed cost per mission	40 RMB/time
c_c	UAV battery swap cost	5 RMB/time
α	Timeout penalty per unit time	3.00 RMB/min
β	Waiting penalty per unit time	1.20 RMB/min
Q_{max}^u	UAV maximum payload	$0.3 Q_{max}^r$
v_{air}	UAV cruising speed	$3 v_r$
c_u	UAV unit energy cost	$0.3 c_r$
T_0	Heuristic initial temperature	100
α_{sa}	Cooling rate	0.95
τ	ng-path neighborhood size	8
I_{ter}	Maximum iterations	2000

5.2. Algorithm Performance Evaluation

5.2.1. Performance Comparison Across Different Instance Scales

As Table 3 shows, we present the average performance comparison between the improved branch-and-price (IBP) algorithm and the benchmark algorithms when solving instances of varying scales. The final results represent the average of 10 independent runs for each instance.

Table 3. Comparative average results on solving different-scale instances.

Instance	IBP						ALNS		Gurobi			
	Num	Gap	Col	Cut	Node	Time	Num	Gap	Num	Gap	Node	Time
C25-1	9	0	92.4	5.8	1.4	56.12	5	3.84	9	0	251.3	415.22
C25-2	8	0	112.5	3.1	1.2	81.34	4	3.55	8	0	228.4	350.18
R25-1	12	0	95.2	7.1	1.4	69.88	5	3.92	12	0	275.1	442.65
R25-2	11	0	114.6	4.8	1.3	112.45	4	5.35	11	0	219.8	458.30
RC25-1	8	0	101.3	5.7	1.5	129.60	4	3.95	8	0	792.4	485.12
RC25-2	8	0	151.2	5.1	1.5	155.70	2	5.72	8	0	1165.2	305.45
C50-1	9	0	255.6	14.2	1.5	472.15	3	3.82	0	11.25	2915.4	1800.00
C50-2	8	0	249.4	16.1	1.7	375.82	3	6.45	0	21.30	3150.2	1800.00
R50-1	12	0	205.3	22.5	2.1	498.50	4	4.12	1	18.15	1982.5	1765.20
R50-2	11	0	198.8	24.3	2.5	735.20	3	5.01	0	26.40	1950.8	1800.00
RC50-1	8	0	175.4	37.1	2.3	502.45	2	3.92	1	21.15	1612.4	1650.35
RC50-2	8	0	171.5	68.2	2.6	458.33	2	5.10	0	9.45	1790.6	1800.00
C100-1	8	3.42	785.4	54.2	2.8	2285.40	0	5.62	-	-	-	-
C100-2	5	2.85	815.6	60.4	3.2	2110.55	1	9.88	-	-	-	-
R100-1	9	4.55	825.1	69.5	2.9	2015.80	1	7.82	-	-	-	-
R100-2	10	3.52	866.3	63.2	2.8	2650.35	0	8.25	-	-	-	-
RC100-1	5	2.78	851.9	95.4	3.1	2365.40	0	6.95	-	-	-	-
RC100-2	7	1.15	745.2	81.3	3.5	1945.20	1	5.95	-	-	-	-

Notes: Num refers to the number of instances where the optimal solution was successfully obtained within the specified time limit. Gap represents the average relative percentage error at algorithm termination, calculated as $Gap = [(UB - LB)/UB] \times 100\%$, where UB denotes the best upper bound, and LB denotes the best lower bound. For the ALNS algorithm, Gap refers to the average deviation between the obtained solution and the lower bound. Col/Cut denotes the average number of columns and cutting inequalities generated during the column generation process. Node is the average number of search nodes in the branch-and-bound tree. Time is the average calculation time in seconds. Num indicates the number of instances where ALNS successfully found a known optimal solution. The symbol “-” indicates that the solver failed to find a feasible solution within the time limit.

The experimental results demonstrate that the IBP algorithm achieves a favorable balance between solution accuracy and computational efficiency. On small-scale instances, both IBP algorithm and Gurobi successfully obtained optimal solutions for all cases; however, IBP exhibited substantially higher computational efficiency. For instance, in the C25-1 case, IBP required 56.12 s, whereas Gurobi needed 415.22 s. This indicates that the heuristic column generation strategy designed in this study effectively accelerates convergence.

As the scale increased to 50 nodes, the solving capability of Gurobi declined sharply, failing to obtain optimal solutions within 1800 s for most instances. In contrast, IBP maintained high stability, successfully finding optimal solutions for all instances with computation times controlled between 400 s and 750 s, thereby validating the algorithm's stability at medium scales. In large-scale instances, Gurobi failed to yield feasible solutions due to memory overflow or timeouts. Although IBP did not achieve full convergence in these cases, the average optimality gap remained between 2.85% and 4.55%, significantly outperforming the ALNS algorithm. Despite the introduction of ng-path relaxation, the state space of the pricing sub-problem remains massive, making it difficult to completely close the lower bound gap in the late stages of column generation. Nevertheless, from an engineering perspective, IBP is capable of providing high-quality dispatching schemes.

5.2.2. IBP Algorithm Performance Analysis

Table 4 further illustrates the internal operational metrics of the IBP algorithm.

Table 4. The performance of the IBP algorithm.

Instance	No. Ins	No. Opt	Solved					Unsolved		
			Avg. Node	AM. Time	AS. Time	Avg. Time	ALP-IP	Max. NGap	Avg. NGap	Min. NGap
C25-1	12	12	1.75	0.12	385.40	580.25	3.95	-	-	-
C25-2	9	9	2.05	0.11	1985.60	2850.45	3.75	-	-	-
R25-1	8	8	1.62	0.08	295.50	472.30	4.02	-	-	-
R25-2	11	9	1.85	0.11	1710.25	2505.40	7.82	13.10	10.25	7.65
RC25-1	8	6	2.15	0.18	2855.40	4050.80	5.92	11.95	7.85	3.80
RC25-2	8	6	2.30	0.14	3680.15	4995.20	7.55	5.10	4.32	3.55
C50-1	12	12	2.15	0.12	1755.40	2910.50	8.65	-	-	-
C50-2	9	9	2.25	0.19	3890.20	5920.60	7.82	-	-	-
R50-1	8	7	1.65	0.11	1695.50	2650.80	7.95	7.85	7.85	7.85
R50-2	11	9	2.58	0.15	4350.60	7215.40	7.15	5.45	4.65	3.82
RC50-1	8	7	2.15	0.23	6280.50	9150.35	5.82	8.25	8.25	8.25
RC50-2	8	6	2.15	0.25	5410.20	8350.40	6.72	13.15	12.05	10.95
C100-1	12	11	2.05	0.14	7150.40	12,850.60	8.15	5.55	5.55	5.55
C100-2	9	6	2.65	0.23	7950.80	11,450.20	7.65	9.10	5.85	4.15
R100-1	8	7	2.55	0.25	5120.30	8210.50	7.35	6.95	6.95	6.95
R100-2	11	6	2.62	0.26	11,450.20	14,560.80	9.85	15.65	8.92	3.25
RC100-1	8	6	2.15	0.35	12,450.50	16,210.30	8.52	14.55	10.95	7.45
RC100-2	8	6	2.35	0.22	11,680.40	13,950.50	6.25	11.55	8.95	6.55

Notes: Instance/No. Ins refers to the instance group name and the total number of test instances in that group. No. Opt indicates the number of instances successfully solved to optimality within the specified time limit. Solved/Unsolved represent the performance indicators for converged and partially converged instances, respectively. Avg. Node is the average number of search nodes in the branch-and-bound tree. AM. Time/AS. Time denote the average cumulative time spent solving the RMP and the pricing sub-problems, respectively, during the column generation process. ALP-IP is the relative deviation between the linear relaxation objective value at the root node and the final integer optimal objective value. NGap is the relative percentage error for unsolved instances at termination, calculated as $Gap = [(UB - LB) / UB] \times 100\%$, where UB denotes the best upper bound, and LB denotes the best lower bound. Max/Avg/Min represents the maximum, average, and minimum values of the corresponding indicators.

By comparing the time spent on the master problem and the sub-problems, it is found that the time required to solve the master problem is extremely short, while the vast majority of computational time is consumed by the pricing sub-problems. As the instance scale increases from $n = 25$ to $n = 100$, the solution time exhibits exponential growth. Nevertheless, the algorithm still converges within a reasonable time, owing primarily to the hybrid solving mechanism designed in this paper, which uses heuristic search to quickly filter paths with negative reduced costs and only invokes the exact labeling algorithm when necessary. Additionally, the ng-path relaxation strategy, by applying strict access restrictions only to nodes within the neighborhood, reduces the state space complexity from $O(n \cdot 2^n)$ to $O(n \cdot 2^T)$, effectively preventing the explosive growth of label quantities and enabling the solution of large-scale sub-problems.

An examination of the branch-and-bound search tree indicates that the number of explored nodes remains remarkably low, regardless of instance scale. This suggests that the lower bound obtained after solving the linear relaxation at the root node is exceptionally tight. The minimal number of nodes also validates the effectiveness of the arc–resource coupling branching strategy. This strategy can partition the solution space quickly through powerful constraints when fractional solutions appear, allowing the algorithm to find integer solutions with very few branches. For unsolved instances at the $n = 100$ scale, both maximum and average optimality gaps remain modest—for example, the average gap for the RC100-2 group is only 8.73%. This indicates that even when computational resources are limited and the branch-and-bound tree cannot be fully traversed, the IBP algorithm can still provide high-quality approximate optimal solutions, demonstrating significant practical utility in engineering contexts. To evaluate the efficacy of the symmetry-breaking strategies, comparative tests were conducted. The results indicate that the number of explored nodes in the search tree was reduced by 30–45%, significantly accelerating convergence.

5.2.3. Model Effectiveness Analysis

To verify the necessity and superiority of the nonlinear energy consumption model we have considered in this paper compared to the traditional linear model, the following comparative experiment was designed using the C50 instance with an environmental wind speed set at $v_w = 8$ m/s. The result is as shown in Table 5.

Table 5. Comparison of schemes of linear model and nonlinear model.

Metric	Linear Energy Model	Nonlinear Energy Model
Total Planned Cost	4120.5 RMB	4358.2 RMB
Number of UAV Tasks	38 orders	32 orders
Mission Success Probability	71.5%	100%
Number of Power-Failure Paths	8 paths	0 paths
Actual Execution Cost	6520.5 RMB	4358.2 RMB

The results indicate that while the linear model yields a lower theoretical cost during the planning stage, it exhibits insufficient robustness. In a strong wind environment, approximately 28.5% of UAV paths become infeasible in actual execution. This is mainly attributed to the model’s neglect of the excessively high energy consumption associated with headwind flight, which ultimately leads to premature battery depletion. In contrast, the nonlinear model ensures 100% feasibility of all paths by proactively avoiding long-distance segments against the wind or reducing the payload per trip.

5.2.4. Iterative Convergence Analysis of the IBP Algorithm

In summary, the numerical experiments validate the superiority of the IBP algorithm in solving the URCDP. Compared to commercial solvers and traditional heuristic algorithms, IBP achieves simultaneous improvements in both solution accuracy and computational efficiency. These enhancements are realized via ng-path relaxation to regulate the state space, a hybrid solution strategy to accelerate subproblem resolution, and an efficient branching rule. Accordingly, the algorithm is especially well-suited for large-scale, multi-constrained instant delivery scheduling scenarios.

Figure 3 presents the algorithm iteration comparison results, intuitively illustrating the convergence trend of the objective function value for the IBP algorithm and comparative algorithms during the iterative process. As shown, the objective value of the IBP algorithm drops sharply in the early iterative stages and then stabilizes, with a convergence speed significantly faster than the ALNS and traditional BP algorithms. This performance stems from the IBP algorithm’s hybrid pricing sub-problem-solving mechanism: heuristic

search rapidly generates high-quality negative reduced cost paths in the early stages to accelerate master problem convergence, while the exact labeling algorithm and ng-path relaxation strategy ensure solution accuracy in the later stages to avoid local optima. In contrast, although the ALNS algorithm initially descends, it maintains a significant deviation from the optimal value after convergence and fails to reach the global solution; the traditional BP algorithm exhibits slow convergence and is prone to stagnation when handling large-scale instances.

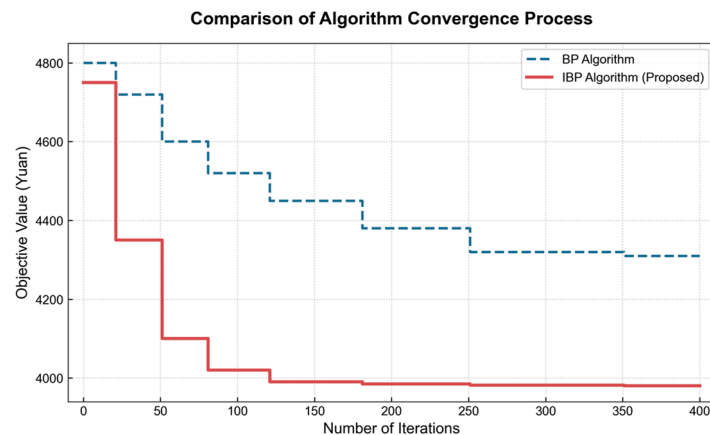


Figure 3. Algorithm iteration comparison chart.

Figure 4 compares the results under different sub-problem solving strategies, showcasing the differences among the complete IBP algorithm (Both), heuristic pricing only (GhOnly), ng-path relaxation only (DngOnly), and traditional branch-and-price (Baseline) across three indicators: the number of unsolved instances, average solving time, and average optimality error.

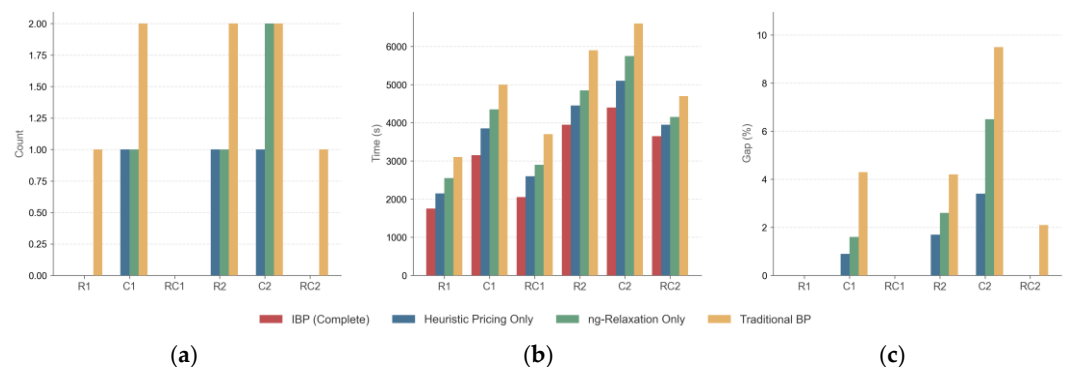


Figure 4. Comparison of solution results under different sub-problem-solving strategies: (a) number of unsolved instances; (b) average computation time; (c) average optimality gap.

Regarding the number of unsolved instances in Figure 4a, the Baseline algorithm performed the worst, failing to converge on 11 out of 56 test instances. The GhOnly and DngOnly strategies reduced the number of unsolved instances to six and seven, respectively, while the IBP (Both) algorithm only had three unsolved instances, demonstrating significantly better convergence than the other strategies. The comparison of average solving time in Figure 4b shows that the average solving time for the IBP (Both) algorithm was 1850 s, representing a 41.57% reduction compared to the 3150 s of the Baseline algorithm, a 13.12% reduction compared to the 2120 s of the GhOnly algorithm, and an 18.35% reduction compared to the 2265 s of the DngOnly algorithm. This fully reflects the collaborative acceleration effect of heuristic search and ng-path relaxation. Finally, the average optimality error comparison in Figure 4c indicates that the IBP (Both) algorithm's error was only

1.57%, far lower than that of the Baseline (9.32%), GhOnly (3.54%), and DngOnly (6.91%) strategies, underscoring a clear advantage in solution quality.

5.3. Key Parameter Sensitivity Analysis

To analyze the impact of core parameters on algorithm performance and total model cost, some key parameters were selected, such as UAV payload capacity (Q_{max}^u), unit energy cost (c_u), timeout penalty coefficient (α), environmental wind speed' (v_w) and number of wind scenarios $|S|$. Sensitivity analysis was conducted on the C50-1 instance, with the results shown in Figure 5.

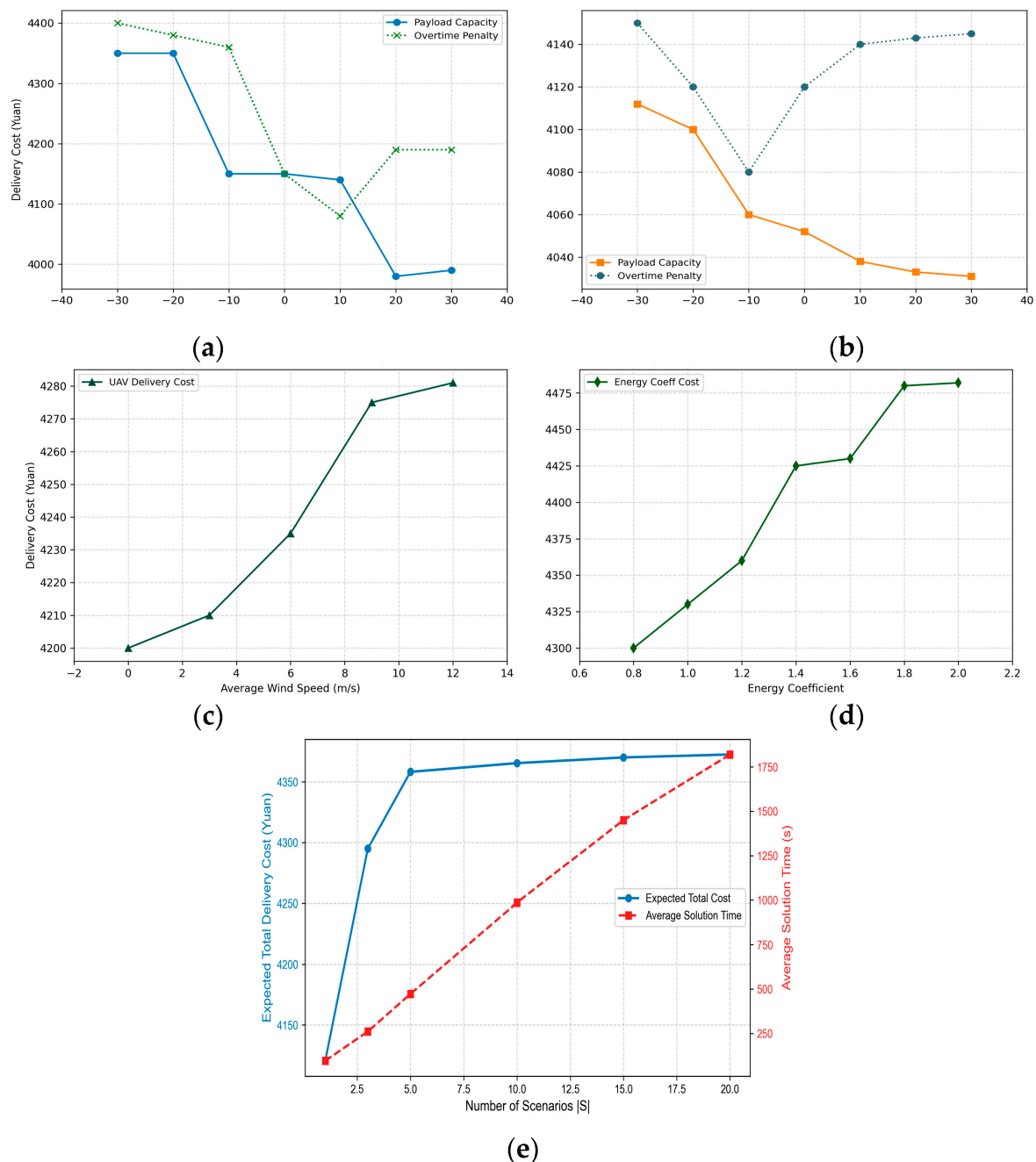


Figure 5. Sensitivity analysis of key parameters: (a) variation in UAV-related parameters; (b) variation in rider-related parameters; (c) impact of environment wind speed; (d) impact of energy consumption coefficient; (e) impact of scenario number on cost and efficiency.

Figure 5a illustrates that the marginal benefit of increasing UAV payload capacity on reducing total costs gradually diminishes. When the payload capacity exceeds 0.3 times that of riders, the additional cost reduction falls below 5%. Figure 5b shows that a 30% increase in unit energy cost leads to a 15.2% rise in total cost, highlighting that reducing energy consumption is a core direction for future optimization. Figure 5c depicts the variation in delivery cost as the average wind speed increases from 0 m/s to 12 m/s. The total cost exhibits a nonlinear increase with wind speed, with the slope of the curve steepening significantly when wind speed exceeds 6 m/s. At wind speeds above 10 m/s, the model automatically reduces UAV utilization, assigning more long-distance orders to riders. Figure 5d shows that as the energy consumption coefficient increases, the collaborative mode effectively mitigates cost pressures by reallocating tasks between riders and UAVs. With a 30% increase in the energy coefficient, the total cost for the collaborative mode rises by only 10.2%, far lower than the 25.8% increase seen in a pure UAV mode. As shown in Figure 5e, increasing the number of stochastic wind scenarios $|S|$ allows the delivery scheme to capture uncertain energy risks more comprehensively, causing expected total costs to rise significantly within the range $|S| \leq 5$; beyond this threshold, the expected cost tends to converge. Further increasing the number of scenarios has a negligible impact on the solution structure but substantially increases the computational burden associated with label expansion in the pricing subproblem. Moreover, the solution time grows linearly with $|S|$. Considering the trade-off between solution robustness and computational efficiency, selecting $|S| = 5$ as the baseline experimental parameter is well justified. In summary, the sensitivity analysis reveals that total costs are most influenced by wind variance and payload weight. Specifically, as wind speed increases beyond 8 m/s, the collaborative framework reallocates 28% of UAV routes to riders to maintain 100% mission feasibility.

5.4. Collaborative Model Advantage Analysis

To quantify the stability of the collaborative scheduling mode, the impact of time-window width variations on the delivery costs of the collaborative mode (DRCPD) and the pure rider mode (R-only) was compared, as shown in Figure 6.

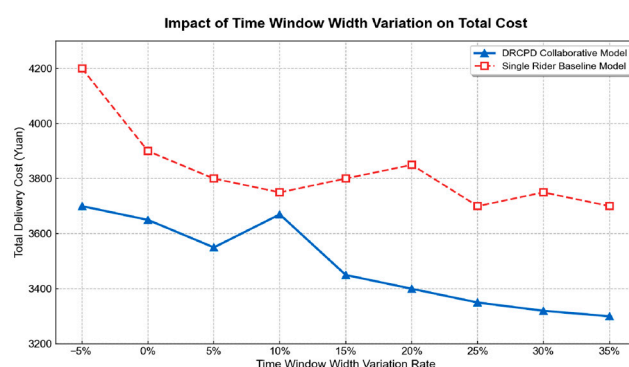


Figure 6. Model stability analysis.

The results demonstrate that as time windows widen, delivery costs decrease in both operational modes; however, the UAV–rider collaborative mode consistently achieves superior economic performance across all test scenarios. Under tight time-window constraints in particular, the collaborative mode leverages the straight-line flight capability of UAVs to effectively avoid significant timeout penalty costs. When the time-window variation rate exceeds 15%, the single-rider mode encounters operational bottlenecks due to ground network constraints, whereas the collaborative mode continues to reduce costs by flexibly adjusting air–ground capacity allocation.

5.5. Statistical Analysis of Scenario Cardinality

To justify the selection of five representative wind scenarios ($|S|=5$) and assess the stability of the stochastic optimization results, a sensitivity analysis combined with one-way ANOVA was conducted. The model was evaluated under $|S| \in \{3, 5, 10, 20\}$, with 20 independent runs per setting to ensure statistical reliability. We evaluated the model's performance across different scenario counts, conducting twenty independent runs for each setting to ensure statistical validity.

As reported in Table 6, increasing the scenario count from 3 to 5 significantly improves the objective value ($p < 0.05$), whereas further increases to 10 and 20 yield no statistically significant improvements relative to $|S|=5$ ($p = 0.425$ and $0.792 > 0.05$), indicating convergence of the optimization results. Although larger scenario sets slightly reduce the standard deviation, they incur a disproportionate increase in computational cost: $|S|=20$ requires over 400% more CPU time than $|S|=5$.

Table 6. Statistical Analysis of Model Performance Under Different Wind Scenario Counts.

$ S $	Mean Objective	Std. Dev.	Avg. CPU Time (s)	Stability Error (%)	p -Value
3	1258.42	12.54	145.2	4.8%	0.008 *
5	1212.15	4.21	388.5	1.2%	-
10	1210.08	2.15	962.4	0.5%	0.435
20	1209.65	1.08	2155.7	0.1%	0.792

Note: * Significant at the $\alpha = 0.05$ level. $|S| = 5$ is the reference group for p -value calculations.

Given the time-critical nature of urban instant delivery, $|S|=5$ achieves an effective balance between statistical accuracy and computational efficiency, providing a stable representation of wind uncertainty while supporting real-time decision making.

6. Discussion and Conclusions

Based on the numerical experiments and comparative analyses conducted, the proposed UAV–rider collaborative dispatching method demonstrates significant practical value for urban instant delivery under complex wind conditions. A primary advantage of the proposed improved branch-and-price (IBP) algorithm is its ability to achieve a strong balance between solution accuracy and computational efficiency for small- to medium-scale instances. By integrating nonlinear aerodynamic energy evaluations directly within the label extension of the pricing sub-problem, the model effectively mitigates linearization errors while maintaining tractability.

However, this study also reveals certain limitations. As observed in extremely large-scale instances, the massive state space generated by the stochastic wind scenarios makes it computationally difficult to completely close the lower bound gap within a reasonable timeframe. Despite this disadvantage, the results consistently validate our primary study objectives. The numerical findings confirm that coupling a scenario-based two-stage stochastic programming framework with dynamic nonlinear energy tracking effectively prevents power failure risks and enhances the overall resilience and feasibility of urban delivery networks.

This paper addresses the challenges of urban instant delivery by constructing a UAV–rider collaborative dispatching model that accounts for stochastic wind conditions and nonlinear energy consumption. An improved branch-and-price (IBP) algorithm, integrating a hybrid pricing strategy and an arc–resource coupling branching strategy, was designed to solve the resulting large-scale stochastic program. Numerical experiments conducted across various dimensions lead to the following conclusions:

- (1) In complex wind scenarios, traditional linear energy models often underestimate power consumption, leading to nearly 30% of UAV missions failing due to energy depletion. In contrast, the proposed aerodynamic-based nonlinear model ensures the operational feasibility of scheduling schemes through precise energy prediction.
- (2) The IBP algorithm significantly outperforms the Gurobi solver and ALNS heuristics in terms of solution quality and efficiency for large-scale instances. By evaluating nonlinear constraints directly within the sub-problems, the algorithm overcomes the computational bottleneck of heterogeneous collaborative scheduling, achieving over 40% higher efficiency than commercial solvers at the 100-node scale.
- (3) Managerial insights suggest that instant delivery platforms should dynamically adjust UAV service radii and payload thresholds during adverse weather. Utilizing riders as a stable base capacity provides a critical risk-hedging mechanism to ensure system reliability under stochastic uncertainty.

While the proposed model effectively handles common wind fluctuations, a practical limitation is that it operates within the UAV's standard safety thresholds. Extreme weather events are not explicitly modeled, as such conditions would necessitate a mandatory suspension of flights rather than tactical routing adjustments. Future research could investigate multi-modal recovery frameworks to manage rare events where UAV operations become entirely unfeasible. Furthermore, integrating real-time dynamic order arrivals and emergency scheduling for hardware failures remains a promising direction for achieving fully adaptive autonomous logistics systems.

Author Contributions: C.S.: research conception, data investigation, modeling and experimental advancement, article writing and polishing. C.Z.: research discussions, original draft and experiments. S.C.: research framework design and guidance, conclusion review and supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (2024YFD2400104).

Data Availability Statement: The datasets presented in this article are not readily available because they are part of an ongoing study with a larger research team. Requests to access the datasets may be directed to one of the authors when the other study is completed.

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

References

1. Wu, J.Q.; Dian, R.; Zi, T.S.; Li, Y.Q. Drone delivery: A systematic review on technology, efficiency, and applications. *J. Transp. Syst. Eng. Inf. Technol.* **2025**, *25*, 34–49. (In Chinese) [[CrossRef](#)]
2. Eskandaripour, H.; Boldsai Khan, E. Last-mile drone delivery: Past, present, and future. *Drones* **2023**, *7*, 77. [[CrossRef](#)]
3. Sadok, A.; Euch, J.; Siarry, P. Vehicle routing with multiple UAVs for the last-mile logistics distribution problem: Hybrid distributed optimization. *Ann. Oper. Res.* **2025**, *351*, 59–99. [[CrossRef](#)]
4. Bai, S.Q.; Li, F.D.; Li, Z.P. Based on improved simulated annealing algorithm of drug distribution unmanned aerial vehicle (uav) route planning. *Intell. Comput. Appl.* **2025**, 1–10. (In Chinese)
5. Li, G.; Yang, J.; Pu, G.; Yang, L. Multi-objective location-allocation optimization of temporary delivery points for medical supplies during large-scale epidemic outbreaks. *Appl. Math. Model.* **2024**, *125*, 721–737.
6. Sheng, T.; Jin, R.; Yang, C.; Qiu, K.; Wang, M.; Shi, J.; Zhang, J.; Gao, Y.; Wu, Q.; Zhou, X.; et al. Unmanned aerial vehicle mediated drug delivery for first aid. *Adv. Mater.* **2023**, *35*, 2208648.
7. Long, Y.; Xu, G.; Zhao, J.; Xie, B.; Fang, M. Dynamic Truck-UAV Collaboration and Integrated Route Planning for Resilient Urban Emergency Response. *IEEE Trans. Eng. Manag.* **2024**, *71*, 9826–9838. [[CrossRef](#)]
8. Yin, Y.; Li, D.; Wang, D.; Ignatius, J.; Cheng, T.; Wang, S. A branch-and-price-and-cut algorithm for the truck-based drone delivery routing problem with time windows. *Eur. J. Oper. Res.* **2023**, *306*, 158–176.
9. Liu, W.S.; Li, W.; Zhou, Q.; Die, X. "Unmanned Aerial Vehicle" Delivery Route Optimization Model and Algorithm. *Transp. Syst. Eng. Inf. Technol.* **2021**, *21*, 176–186. (In Chinese) [[CrossRef](#)]

10. Sun, X.; Fang, M.; Guo, S.; Hu, Y. UAV–rider coordinated dispatching for the on-demand delivery service provider. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *186*, 103571.
11. Niu, G.D.; Wang, X.X. Optimization of Instant Delivery Routing Based on Drone—Rider Collaboration Mode. *Mod. Advert.* **2025**, 10–16. (In Chinese)
12. Chen, J.; Wan, P.; Xu, G. Cooperative Learning-Based Joint UAV and Human Rider Scheduling for Emergency Medical Delivery Service. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 935–949. [\[CrossRef\]](#)
13. Yuan, Q.; Li, R.; Zhou, B.; Lu, J.; Hu, M.; Lai, P.; Zhao, Y.; Zhang, X. Collaborative Truck-UAV Delivery Routing Optimization Under Dynamic Weather Conditions and Customer Demands. *IEEE Trans. Consum. Electron.* **2025**, *71*, 10950–10964. [\[CrossRef\]](#)
14. Johnson, A.M.; Cunningham, C.J.; Arnold, E.; Rosamond, W.D.; Zègre-Hemsey, J.K. Impact of Using Drones in Emergency Medicine: What Does the Future Hold? *Open Access Emerg. Med.* **2021**, *13*, 487–498. [\[CrossRef\]](#)
15. Liu, X.; Yuan, Q.; Wang, G.; Bian, Y.; Xu, F.; Chen, Y. Drones Delivering Automated External Defibrillators: A New Strategy to Improve the Prognosis of Out-of-hospital Cardiac Arrest. *Resuscitation* **2023**, *182*, 109669. [\[CrossRef\]](#) [\[PubMed\]](#)
16. De Silvestri, S.; Capasso, P.J.; Gargiulo, A.; Molinari, S.; Sanna, A. Challenges for The Routine Application of Drones in Healthcare: A Scoping Review. *Drones* **2023**, *7*, 685. [\[CrossRef\]](#)
17. Zègre-Hemsey, J.K.; Cheskes, S.; Johnson, A.M.; Rosamond, W.D.; Cunningham, C.J.; Arnold, E.; Schierbeck, S.; Claesson, A. Challenges & barriers for real-time integration of drones in emergency cardiac care: Lessons from the United States, Sweden, & Canada. *Resusc. Plus* **2024**, *17*, 100554. [\[CrossRef\]](#)
18. Hanafiah, M.H.M.; Al-Humairi, S.N.S.; Taib, M.A.A.M.; Jaharadak, A.A. Analysis and Optimization of Care-purpose Drone for Medical-Based Applications. In *Proceedings of the 2024 20th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6. [\[CrossRef\]](#)
19. Hemalatha, D.; Dharshini, S.; Ramesh, P.; Priya, S. AI Enabled MED Drone for Healthcare Application. *Int. J. Innov. Res. Adv. Eng.* **2024**, *2*, 190–193. [\[CrossRef\]](#)
20. Kaduskar, S.; Parab, P.; Chaudhari, D.; Trivedi, P.; Khan, U.; Menon, C. SkyMedic: The Lifesaving Autonomous Ducted UAV. In *Proceedings of the 2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*; IEEE: Piscataway, NJ, USA, 2024; pp. 789–794. [\[CrossRef\]](#)
21. Güzey, A.; Satman, M. Collaborative Truck/Drone Routing Problem: An Application to Disaster Logistics. *J. Stat. Appl. Sci.* **2024**, *9*, 77–94. [\[CrossRef\]](#)
22. Campbell, H.A.; Bosiocic, V.; Hvala, A.; Brady, M.; Campbell, M.A.; Skelton, K.; Luiz, O. Emerging Research Topics in Drone Healthcare Delivery. *Drones* **2024**, *8*, 258. [\[CrossRef\]](#)
23. Bhattacharya, S.; Singh, A.; Tripathi, S.; Kushwaha, P.; Mondal, H.; Ramanathan, V.; Sikha, D.; Mudgal, S.K.; Bhattacharjee, S.; Bhattacharya, S.; et al. A Systematic Review of Drone Customization and Application in Public Health Innovation. *Cureus* **2025**, *17*, e85435. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Di Puglia Pugliese, L.; Guerriero, F.; Scutellà, M.G. The Last-Mile Delivery Process with Trucks and Drones Under Uncertain Energy Consumption. *J. Optim. Theory Appl.* **2021**, *191*, 31–67. [\[CrossRef\]](#)
25. Lu, X.; Wu, Q.; Zhou, K. Path optimization for UAV food delivery under extreme weather conditions with wind disturbance consideration. *Comput. Ind. Eng.* **2025**, *209*, 111452. [\[CrossRef\]](#)
26. Wu, K.; Lu, S.; Chen, H.; Feng, M.; Lu, Z. An energy-efficient logistic drone routing method considering dynamic drone speed and payload. *Sustainability* **2024**, *16*, 4995. [\[CrossRef\]](#)
27. Zhang, Z.; Zhou, C.; Sheng, L.; Cao, S. Optimization Schemes for UAV Data Collection with LoRa 2.4 GHz Technology in Remote Areas without Infrastructure. *Drones* **2022**, *6*, 173. [\[CrossRef\]](#)
28. Huang, R.; Zhao, X.; Wang, J.Y. Intermodal transportation network design optimization considering demand uncertainty under “dual carbon” background. *J. Transp. Syst. Eng. Inf. Technol.* **2025**, *25*, 1–12. (In Chinese) [\[CrossRef\]](#)
29. Dodge, M.; Mirhassani, S.A.; Hooshmand, F. Modeling wind-affected drone routing under uncertainty. *Expert Syst. Appl.* **2024**, *240*, 122305. [\[CrossRef\]](#)
30. Li, Y.; Liu, M.; Hu, H.; Jiang, X. A cost analysis of the collaborative delivery mode of drones and riders in food delivery scenarios. *Int. J. Logist. Res. Appl.* **2025**, 1–30. [\[CrossRef\]](#)
31. Gao, J.; Wang, Q.; Zhang, X.; Shi, J.; Zhao, X.; Liang, Y.; Guo, B.; Han, Q.; Pan, Y. Cooperative Air-Ground Instant Delivery by UAVs and Crowdsourced Taxis: Joint UAV Station Deployment and Delivery Scheduling. *IEEE Trans. Mob. Comput.* **2025**. [\[CrossRef\]](#)
32. Rema, C.; Costa, P.; Silva, M.; Pires, E.J.S. Task scheduling with mobile robots—A systematic literature review. *Robotics* **2025**, *14*, 75. [\[CrossRef\]](#)
33. Sabuncu, Ö.; Bilgehan, B. Nonlinear energy modeling for UAVs in critical missions using multiplicative calculus. *Sustain. Comput. Inform. Syst.* **2025**, *48*, 101206. [\[CrossRef\]](#)
34. Cheng, C.; Adulyasak, Y.; Rousseau, L.M. Robust Drone Delivery with Weather Information. *Manuf. Serv. Oper. Manag.* **2024**, *26*, 1402–1421. [\[CrossRef\]](#)

35. Lyu, M.; Liu, W.; He, Q.; Wu, L. Urban food delivery service optimisation with coordinated delivery riders and drones. *Transp. Res. Part E Logist. Transp. Rev.* **2025**, *204*, 104412. [[CrossRef](#)]
36. Lu, F.; Liu, J.; Bi, H. Drone–Rider Joint Delivery Routing with Arc Obstacle Avoidance. *Appl. Sci.* **2025**, *15*, 11469. [[CrossRef](#)]
37. Su, X.X.; Yi, T.G.; Qin, H. The branch pricing cut plane method for solving the vehicle routing problem with time Windows and manpower allocation. *J. Transp. Eng. Inf. Technol.* **2021**, *19*, 75–86. (In Chinese) [[CrossRef](#)]
38. Schneider, M.; Stenger, A.; Goeke, D. The electric vehicle-routing problem with time windows and recharging stations. *Transp. Sci.* **2014**, *48*, 500–520. [[CrossRef](#)]

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