Transportation Research Part B 209 (2026) 103469

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

Joint fleet scheduling and cargo flow allocation for air cargo services

Ling Zhu [a], Simon Belieres [b], Mike Hewitt [c], Lingxiao Wu [a,*]

[a] Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, China
[b] Department of Information, Operations and Decision Sciences, TBS Business School, Toulouse, France
[c] Department of Information Systems and Supply Chain Management, Quinlan School of Business, Loyola University Chicago, Chicago, USA

ARTICLE INFO

Keywords:
Air cargo routing
Fleet scheduling
Through cargo connection
Column generation
Integer programming

ABSTRACT

We propose an integrated optimization model designed for the air cargo service network scheduling problem, focusing on next-day delivery. Our model effectively combines dedicated cargo aircraft with belly capacity from passenger airlines to jointly determine optimal cargo flight schedules, cargo fleet routes, and cargo routes. To enhance the efficiency of the air cargo service network, we incorporate through cargo connections into the cargo routing problem. These connections occur when the same aircraft operates consecutive flights within a cargo route. A tailored column-generation-based heuristic is developed to solve the problem. We conduct an extensive set of experiments to validate the performance of our proposed model and algorithm based on the data from our industry partner company. The computational results demonstrate that for small-scale instances, our algorithm rapidly identifies near-optimal solutions. Meanwhile, for medium- and large-scale instances, it consistently delivers superior solution quality compared to the commercial solver. Ultimately, our study highlights that the inclusion of through cargo connections not only enhances the overall service level of the network but also leads to increases in flight load factors and reductions in operational costs.

1. Introduction

Air transportation plays a crucial role in the freight market due to its speed, reliability, and security, particularly in the context of the rapid growth of e-commerce. In 2019, e-commerce represented a global market of US$ 3.5 trillion, accounting for 14% of total retail sales (Boeing 2020). Driven by this growth, air transport has become increasingly well-suited to meet the rising demands of the market, with e-commerce accounting for up to 15% of air cargo volumes by December 2019 (IATA 2020). Meanwhile, customer expectations for faster delivery times and highly streamlined processes have risen significantly. In response, some air cargo carriers, such as Amazon Air and SF Express, have started operating their own fleets of cargo aircraft. While air freight transportation operations rely on dedicated cargo aircraft, they can also depend significantly on the belly space of passenger planes. According to Boeing, cargo transported via passenger flights occupies more than half of the global air cargo capacity (Boeing 2020).

Carriers typically offer a range of fast delivery services with a maximum timeframe of about one week. The option with fastest delivery service is next-day delivery, guaranteeing delivery by 10:30 a.m. (Liu et al. 2019). Consequently, air cargo carriers face minimal time between order receipt and the commencement of transportation, necessitating swift operational adaptations to effectively

* Corresponding author.
E-mail addresses: ling01.zhu@connect.polyu.hk (L. Zhu), s.belieres@tbs-education.fr (S. Belieres), mhewitt3@luc.edu (M. Hewitt), lingxiao-leo.wu@polyu.edu.hk (L. Wu).

https://doi.org/10.1016/j.trb.2026.103469
Received 29 March 2025; Received in revised form 3 March 2026; Accepted 31 March 2026
Available online 7 April 2026

meet these delivery timelines. However, the decision-making process for air cargo carriers is complex, as it involves the coordination of multiple interdependent factors, including deticated fleet routing, cargo flight scheduling and cargo allocation. Given the tight timelines within which these decisions must be made, the traditional methods of decision-making may prove insufficient. Therefore, the implementation of automated optimization-based methods for decisions becomes essential.

For cargo fleet operations decisions, they need to satisfy a wide range of operational constraints. First, cargo fleet operations are subject to legal and regulatory constraints. For example, in China, cargo aircraft can only operate at night (Yıldız and Savelsbergh 2022). In addition, flow conservation for the cargo fleet requires that the number of aircraft of a given type at an airport at the end of the planning horizon must be the same as the number of aircraft of that type at the beginning of the planning horizon (Desaulniers et al. 1997, Yıldız and Savelsbergh 2022). This requirement can ensure the establishment of a periodic flight schedule. Besides, a cargo aircraft route, which consists of a sequence of cargo flights, is typically required to start from and terminate at hub airports (Derigs et al. 2009). Ground operations also represent a significant constraint in the scheduling of cargo aircraft routes, as they necessitate the allocation of adequate turnaround time between adjacent flights within aircraft routes for cargo handling. When planning aircraft routes and schedules, it is sufficient to ensure that turnaround time constraints are met for consecutive flights operated by the same aircraft.

As for decisions on cargo routing, each cargo booking specifies an (origin, destination) (O, D) pair with a delivery time window. Carriers need to design routes to transport cargo within this window, even when cargo is split across multiple cargo routes to optimize capacity utilization. Cargo can be transported by dedicated cargo aircraft and belly capacity of externally scheduled passenger flights booked from passenger airlines. A cargo route can consist of multiple cargo flights or a single flight (either cargo flight or passenger flight), which need to meet the capacity and scheduling needs for flights. However, when consecutive cargo flights on a cargo route are operated by different cargo aircraft, the cargo must be unloaded from one aircraft and reloaded onto another. These transshipment operations introduce additional handling time and require specialized equipment, which is generally available only at hub airports (Xiao et al. 2022, Huang et al. 2023). Reducing the number of transshipment operations is highly desirable, not only because these operations are costly but also because the handling time required for these operations may limit the demand served on time. This is particularly true when serving next-day delivery demand.

Transporting cargo on consecutive flights using the same aircraft eliminates the need for transshipment operations, alleviating the requirement for transshipment to occur at hub airports and reducing the handling time. This type of connections is referred to as a through cargo connection (Xiao et al. 2022). In this paper, we incorporate these connections into our daily air cargo routing problem, highlighting the interdependent relationship between aircraft routing and specific cargo routes. Furthermore, we exclude considerations of cargo loading problem, assuming that cargoes are loaded in the appropriate positions according to their destinations, following the sequence of airports traversed by the aircraft. By incorporating these features and assumptions, we explore the air cargo service network scheduling problem, emphasizing the comprehensive optimization of dedicated cargo fleet operations alongside cargo allocation.

Our main contributions are threefold. First, we propose an integrated optimization model for the air cargo service network scheduling problem. Our model simultaneously coordinates dedicated cargo fleet operations (i.e., flight scheduling, aircraft routing, and fleet assignment), belly capacity booking, and synchronized cargo allocation, providing a decision-making tool to handle the complex interdependency in air cargo logistics.

Second, a key feature of our model is the integration of through cargo connections into the cargo transshipment process to enhance network connectivity. We conduct an extensive computational analysis, based on operational data from our industry partner company, to quantify the impact of these connections. Our results demonstrate that this integration allows the system to serve significantly more demand at lower operational costs. This finding constitutes our second contribution, demonstrating the value of through cargo connections in optimizing air cargo operations.

Third, to efficiently address the integrated optimization model, we propose a tailored column-generation-based heuristic adapted to price out multiple columns per iteration. In the restricted master problem, we introduce a tightened constraint on the cargo flow variables to reduce computational time. We then transform the pricing problem into a longest path finding problem and develop a Δ-longest path algorithm to solve it effectively. After obtaining the linear programming solution, we subsequently solve a mixed-integer programming model using the aircraft routes derived from this process to obtain integer solutions and propose several strategies to improve solution quality. Furthermore, we exploit the totally unimodular property of the model, which holds when the binary variables are fixed. This allows us to relax specific integer variables to continuous ones without compromising solution quality, thereby reducing the solving time.

The remainder of this paper is structured as follows. Section 2 presents a comprehensive review of relevant literature related to our research. Following this, we provide a detailed description of the air cargo service network scheduling problem in Section 3, along with the corresponding integrated optimization model proposed in Section 4. The methodologies for solving this model are delineated in Section 5. Section 6 provides an extensive series of experiments aimed at assessing our algorithm performance and drawing managerial insights. Finally, the key conclusions derived from the study are presented in Section 7.

2. Literature review

This section reviews the literature relevant to our study and is divided into three parts. The first and second parts focus on articles that individually address the two main components of our problem, namely, fleet scheduling and air cargo routing. The fleet scheduling problem can be defined as an integration of the flight scheduling problem, the fleet assignment problem, and the aircraft routing problem. The third part reviews articles that aim to integrate fleet scheduling and air cargo routing.

2.1. Fleet scheduling problems

The literature includes a variety of optimization problems that assist the decision-making related to fleet scheduling. Most of these problems find applications in the context of passenger transportation. In addition to fleet scheduling, crew scheduling is also a critical component of the passenger transportation scheduling problem (Barnhart and Cohn 2004). In this study, we do not address crew scheduling problem, as cargo aircraft are typically operated by two pilots without the requirement for additional cabin crew (Derigs and Friederichs 2013). Fleet scheduling involves multiple planning processes, the main elements of which are (i) flight scheduling, (ii) fleet assignment, and (iii) aircraft routing. More specifically, flight scheduling decisions focus on the selection and timing of the flights, fleet assignment aims to assign aircraft of different types to flight services, and aircraft routing focuses on designing the individual routes followed by the aircraft along the network (Abara 1989, Lohatepanont and Barnhart 2004, Sherali et al. 2006, Zhou et al. 2020). Note that fleet assignment and aircraft routing decisions are often integrated into a unified problem known as the fleet routing problem (Yan and Young 1996, Yan and Tseng 2002).

The optimization process for passenger flight scheduling typically comprises two core stages: schedule construction and schedule evaluation. During the first stage, an initial flight schedule is devised considering factors such as projected demand and airport capacities, while the subsequent phase assesses the feasibility, cost-effectiveness, and performance of the schedule. Both phases are operated iteratively until the schedule meets the satisfaction criterion (Etschmaier and Mathaisel 1985, Yan et al. 2008).

The fleet routing problem is typically modeled by three primary methods: string-based, connection-based, and time-space networks. Barnhart et al. (1998) first introduce the concept of a flight string, that is, a sequence of successive flights that satisfy the requirements for spatial and temporal connectivity. The string-based formulation has been widely adopted for its ability to handle complex operational constraints. Birolini and Jacquillat (2023) utilize this formulation to address the day-ahead aircraft routing problem, integrating predictive analytics for primary delay estimation. Abara (1989) put forth a connection-based model to address the fleet routing problem. This network consists of three types of nodes: virtual sources, virtual sinks, and individual flights. It also consists of three types of arcs: source arcs, termination arcs, and connection arcs. Source and termination arcs connect individual flights to sources and sinks, representing the start or end of a route. Connection arcs are defined for pairs of individual flights that meet spatial and temporal connectivity requirements. The traditional string-based and connection-based methods have been commonly used to model aircraft maintenance routing problems and aircraft recovery models (Al-Thani et al. 2016, Bulbul and Kasimbeyli 2021, Khaled et al. 2018, Safaei and Jardine 2018, Wen et al. 2022). However, these techniques do not account for temporal attributes, which are essential in our air cargo service network scheduling problem. Therefore, we model the air cargo service problem via time-space networks that inherently include time. Berge and Hopperstad (1993) first use time-space networks to address the fleet routing problem. In this framework, nodes represent two-dimensional spatial-temporal coordinates defined by the airport locations, flight departures, and arrival times.

On the other hand, recent studies from the literature aim to integrate these problems to propose a more comprehensive optimization model, striving to achieve synergistic optimization in terms of network efficiency and fleet utilization. Yan and Young (1996) devise a framework utilizing a multi-fleet time-space network to assist carriers in modifying their flight schedules and fleet routes in response to forthcoming changes in market demand conditions. Expanding on this foundation, Yan and Tseng (2002) present an integrated model for fleet route planning and flight scheduling, reducing manual involvement. Yan et al. (2007) extend the fleet scheduling problem to account for market share competition. They incorporate passenger choice behaviors with the aim of reducing fleet expenses as well as passenger expenses. Papadakos (2009) analyzes an integrated airline passenger scheduling problem combining fleet assignment, aircraft routing, and crew pairing, and proposes a Benders decomposition algorithm as well as acceleration strategies. Sherali et al. (2013) introduce an integrated optimization model for the fleet scheduling problem incorporating multiple fare classes, maintenance considerations, and demand recapture to maximize profit. Jamili (2017) integrates aircraft routing, flight scheduling, and fleet assignment into a mixed-integer mathematical model. He also introduces a new robust approach to address uncertainties that relate to travel times. Kenan et al. (2018) propose a two-stage stochastic mixed-integer programming model that incorporates flight scheduling, fleet assignment, and aircraft routing, while addressing demand uncertainty, delays, and deadhead flights. Wei et al. (2020) incorporate endogenous passenger choice represented by a sales-based linear programming approach into an integrated flight scheduling and fleet assignment optimization model to determine the timetable from scratch. Accounting for the propagated delay, Xu et al. (2021) investigate the robust fleet scheduling problem involving flight scheduling, fleet assignment, and aircraft routing.

The existing literature on fleet scheduling focuses primarily on passenger networks, with limited attention given to air cargo networks. However, the significant differences between passenger and air cargo transportation necessitate distinct considerations within the fleet scheduling problems (Roelen et al. 2000, Xiao et al. 2022). First, passenger transportation is schedule-driven, prioritizing individual preferences, whereas air cargo is demand-driven, focusing on delivery deadlines. Since air cargo shipments are indifferent to specific itineraries, carriers possess the flexibility to restructure flight schedules in the short term to match aggregated demand, a practice less feasible in passenger transportation. Second, operational time windows are distinct. Dedicated cargo fleet often operate during the night to avoid conflicts with daytime passenger transportation. Third, cargo transfers between aircraft are restricted to hub airports, a restriction that is not relevant to passengers. Therefore, we propose an optimization model specifically designed for the characteristics of air cargo services, aiming to enhance operational efficiency and improve service levels.

2.2. Air cargo routing problems

Air cargo routing problems aim to optimize the transportation of cargoes over an air cargo service network, which may either be given or treated as an integral part of the network design problem, while respecting capacity constraints. Armacost et al. (2002) address the problem of designing service networks for fleet assignments as well as package routes. Li et al. (2012) propose a mixed-integer programming model to allocate shipments to flights considering piecewise linear cargo transportation prices. Azadian et al. (2012) introduce a Markov decision process model for dynamically routing time-sensitive air cargo, utilizing the departure delay estimation developed based on experience and real-time information. Lee et al. (2019) develop an integrated model that locates hubs and distributes cargoes through routes, which are limited to a maximum of two transshipment operations.

While the aforementioned papers address air cargo transportation decision-making, they solely involve dedicated cargo aircraft. On the other hand, it is acknowledged that air cargo transportation companies rely heavily on the belly capacity rented from passenger airlines. Considering the simultaneous utilization of cargo aircraft and rental belly capacity, Yu et al. (2017) present a bilevel model to describe the air cargo network in which the upper model determines hub locations while the lower model optimizes cargo routes. Subsequently, Yu and Jiang (2024) investigate the integrated air-rail network that includes cargo aircraft, passenger aircraft bellies, and high-speed train bellies, and construct a bi-level optimization model, where the upper model designs the network, and the lower model optimizes cargo routing.

2.3. Integrated optimization of fleet scheduling and air cargo routing

To enhance operational efficiency within air cargo service networks, researchers focus increasingly on the interactions between fleet scheduling and cargo routing. Yan et al. (2006) utilize cargo-flow and fleet-flow networks to construct an integrated scheduling model that encompasses airport selection, fleet scheduling, and cargo routing, with the aim of maximizing operational profit. Subsequently, Tang et al. (2008) extend this model by integrating passenger flights, cargo flights, and combined flights into the network. Li et al. (2007) propose an integrated optimization model and a Benders decomposition-based approach to address simultaneously fleet assignment and cargo routing problems. Derigs et al. (2009) develop two models that integrate flight selection, aircraft rotation planning, and cargo routing, while considering a homogeneous fleet of aircraft. Derigs and Friederichs (2013) further explore the fleet assignment problem and present a joint optimization framework that leverages existing flight schedules to decide flight selection, plane rotation planning, and freight routing. Yıldız and Savelsbergh (2022) develop an optimization model to determine flight schedules, aircraft routes, and cargo routes within a single-flight shipment network using self-owned cargo aircraft, with rental belly capacity and ground transportation to meet coverage limits. Xiao et al. (2022) introduce the concepts of through cargo connections and short through cargo connections and propose an integrated optimization model to maximize revenue and minimize transportation costs for aircraft tail assignment and cargo routing problems. Zheng et al. (2023) investigate the duration of cargo stay at the air cargo hub and develop a model for optimizing network planning and scheduling that identifies the optimal hub location, flight schedule, fleet deployment, and cargo routing.

Works in the existing literature mainly integrate air cargo routing problem with a part of subproblems of fleet scheduling. Xiao et al. (2022) integrate aircraft routing, fleet assignments, and cargo routing assuming that flight schedules are known. Considering the operational flexibility of the network, we extend the work of Xiao et al. (2022) by simultaneously determining the flight schedules while designing operations. Additionally, we integrate flow conservation constraints for each aircraft type at hub airports to optimize fleet management. Zheng et al. (2023) jointly optimize flight scheduling, flight deployment, and cargo routing. In contrast to their approaches, we put forth a more comprehensive optimization model that integrates the optimization of these three interrelated subproblems, considering their coupling relationship. In addition, we incorporate belly capacity into our air cargo transportation framework to enhance the overall effectiveness of the air cargo service network.

To the best of our knowledge, the most closely related work that integrates fleet scheduling and cargo routing is that of Yıldız and Savelsbergh (2022). However, we note two significant differences between their approach and ours. First, although they permit multimodal transportation, such as ground-air-ground combinations with optional ground legs, we explicitly exclude ground transportation due to its prohibitive 5.5-hour modal transfer time between ground transportation and air transportation. Instead, our focus is on a pure air cargo service network tailored for time-sensitive morning deliveries. Besides, Yıldız and Savelsbergh (2022) assume that when cargo is transported by air, it is moved exclusively by a single direct flight, either using the cargo aircraft or passenger aircraft. As such, they disregard air cargo routes involving multiple flight legs and possible transshipment operations via dedicated cargo aircraft.

Second, they treat all airports as hub airports, allowing aircraft to initiate and terminate flights anywhere. In practice, however, only a small number of airports function as true hub airports. The concentration of fleet operations at these selected hubs restricts the connectivity of the air cargo service network. Indeed, as transshipments are solely performed at hub airports with specialized equipment, the need for flights tends to be greater through these locations. In next-day delivery systems, the number of flights an aircraft can operate is limited, which further restricts network connectivity. In this paper, we build upon this work by integrating cargo transshipment, including through cargo connections within the air cargo routing problem, while also considering the constraint of a limited number of hub airports.

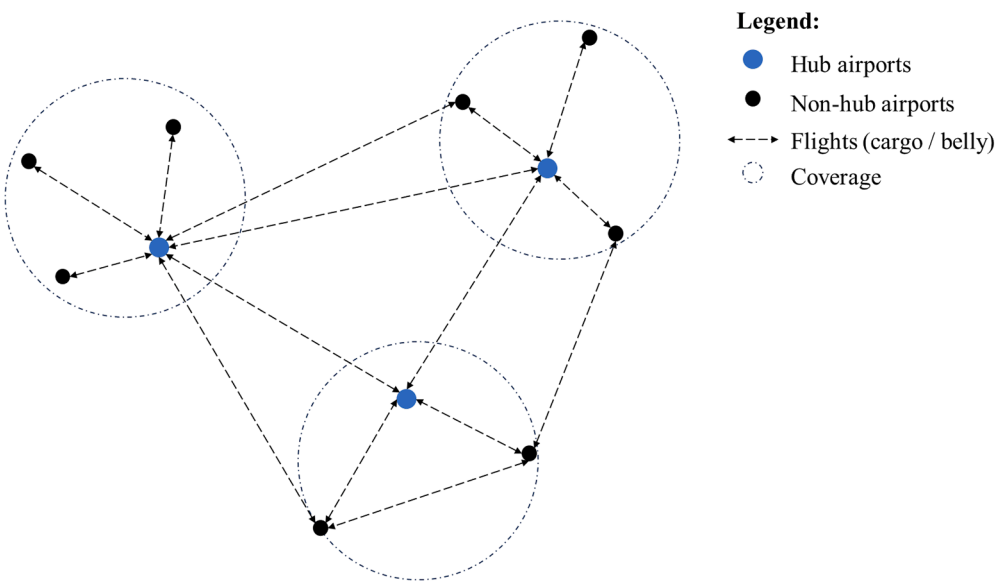

Fig. 1. Illustration of air cargo service network.

3. Problem description

In this section we provide a detailed description of the considered air cargo service network scheduling problem faced by air cargo carriers, which aims to reduce overall operational costs. Two resources are used for air cargo transportation. The first is dedicated cargo aircraft owned by the carriers. The second is passenger flight belly capacity leased from passenger airlines. The hub-and-spoke network structure and the point-to-point network structure are two prevalent configurations in air transportation network, applicable to both passenger airline networks and air cargo networks. This study focuses on the service network scheduling problem rather than the network structures. Therefore, we presume that both passenger aircraft and dedicated cargo aircraft operate within an air transportation network that integrates features of both “point-to-point” and “hub-and-spoke” features, allowing for greater flexibility and efficiency in scheduling and routing, as depicted in Fig. 1.

Considering this air cargo service network, air cargo carriers are tasked with planning the routes and schedules of their own cargo fleet. This planning process encompasses three key problems: the flight scheduling problem, the fleet assignment problem, and the aircraft routing problem. The flight scheduling problem involves determining schedules for cargo flights considering factors such as time windows for cargo demand, available operational times for cargo aircraft, and time constraints for connecting flights. The fleet assignment problem focuses on deciding the type of aircraft to operate each cargo flight. Different types of aircraft have varying capacities and operational costs per flight operation, making it crucial to assign the most suitable aircraft type to each flight to optimize capacity utilization and minimize costs. The aircraft routing problem involves determining the sequence of flights that each aircraft will operate, ensuring that the destination airport of one flight matches the origin airport of the subsequent flight, with the connection time meeting the turnaround time constraints. Besides, within the overnight operational time window, each aircraft route must begin and end at hub airports. Specifically, the origin of the first flight and the destination of the last flight in the sequence must be hub airports, which may be the same or different.

Collectively, flight schedules, fleet assignments, and aircraft routes induce a service network through which cargo may be transported from its origin to its destination. Further, complementing the capacity in that network from dedicated flights is the belly capacity of passenger flights. We assume that the available capacities and schedules of such passenger flights are known to the planner when routing cargo. We focus on next-day services, which guarantee that cargo is routed to arrive at its destination by the following morning.

Furthermore, the problem addressed in this study is positioned at the tactical short-term planning level for air cargo service network scheduling problem, with the objective of constructing a cyclical flight schedule and fleet routing plan from scratch. We assume a stable air cargo demand profile for a short-term period, which represents a reliable, aggregated forecast for a typical operational period. Consequently, the problem is designed to generate a self-sustaining baseline flight schedule that can be executed repeatedly. While day-to-day operational volatility (e.g., demand spikes or weather disruptions) inevitably occurs, such deviations are typically managed through separate disruption management protocols (e.g., ad-hoc charters or spot market adjustments). Therefore, we treat the air cargo demand in our study as known and fixed for the purpose of optimizing the core service network scheduling. Associated with a cargo demand is an origin airport, a destination airport, the earliest time the cargo can be picked up at its origin and the latest time at which it can arrive at its destination.

Air cargo can be shipped with or without transshipment (Zheng et al. 2023). When transshipment is involved, the transfer from one aircraft to another must take place at hub airports (Li et al. 2007, Xiao et al. 2022, Huang et al. 2023). Therefore, we consider

Table 1
Comparison of standard cargo connections and through cargo connections.

Connections	Must be operated by the same aircraft	Must happen at hub airports	Minimum transshipment time
Standard cargo connection		✓	Standard cargo transshipment time
Through cargo connection	✓		Turnaround time for aircraft

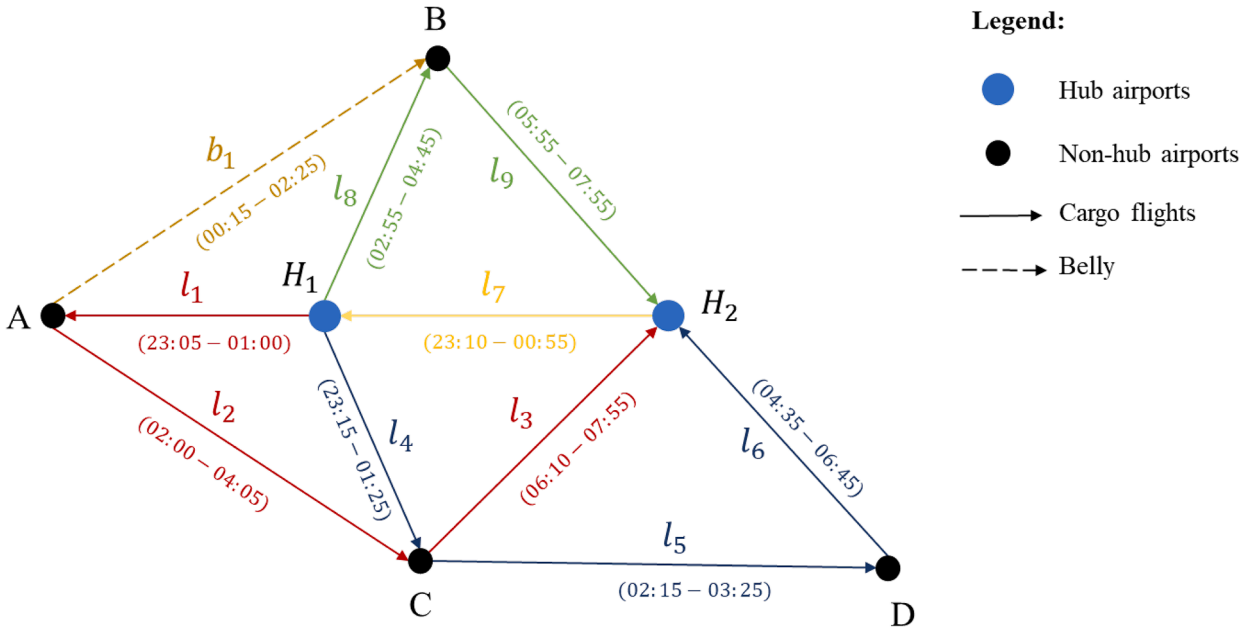

Fig. 2. Illustration of cargo routes.

two types of cargo routes: (i) direct flights, where cargo is shipped from origin to destination via a single leg, either by a dedicated aircraft or belly capacity, and (ii) transshipment routes, where cargo is transported through multiple legs by a dedicated aircraft with intermediate stops between origin and destination. Direct routes involve a single cargo or passenger flight from the origin to the destination airport. In contrast, transshipment routes consist of a series of cargo flights.

Transshipment routes can be operated by the same aircraft or different aircraft. When different aircraft are used, the connecting airport must be a hub airport, and the connection time must meet the standard cargo transshipment time. However, when the same aircraft operates the connecting flights, this is termed a through cargo connection (Xiao et al. 2022). In such cases, the connecting airport can be a hub or non-hub airport, and the connection time can be shorter than the standard transshipment time, provided it meets the turnaround time for the aircraft. Table 1 summarizes the essential features of the two types of cargo connections: standard cargo connections, and through cargo connections.

To illustrate this problem, consider the simple physical network illustrated in Fig. 2, which includes two hub airports, H_1 and H_2, along with four non-hub airports: A, B, C, and D. The fleet within this network consists of four aircraft. Taking into account a turnaround time of 45 minutes, we can generate aircraft routes for each aircraft that satisfy the previously described constraints. In this figure, cargo flights of the same color indicate operations by the same aircraft, such as $l_1 - l_2 - l_3$. The flight schedule is annotated on each directed edge, which specifies the corresponding departure and arrival times. For instance, edge l_7 (23:10-00:55) represents a cargo flight l_7 operating from airport H_2 to H_1, with scheduled departure at 23:10 and subsequent arrival at 00:55 the following day. In addition, there is a belly resource connecting airports A and B, which operates on a specific schedule obtained from the passenger airline. Assuming a standard cargo transshipment time of two hours, and each (O,D) pair has a cargo demand with a specified time window from 23:00 to 8:00, some of generated cargo routes can be described as follows.

(i) For the (O,D) pair $H_2 - B$, there is a single feasible cargo route, which involves flights $l_7 - l_8$. As these flights are operated by different aircraft, this route involves a standard cargo connection, and thus a transshipment operation.

(ii) For the (O,D) pair $H_1 - C$, there are two feasible cargo routes. The first route involves flights $l_1 - l_2$. Because these flights are operated by the same aircraft, and as the stay time at airport A is greater than 45 minutes, the connection is referred to as a through cargo connection. The second route is direct and involves flight l_4.

(iii) For the (O,D) pair $A - B$, there is a single feasible cargo route, which involves the belly capacity b_1.

(iv) For the (O,D) pair $A - D$, there is no feasible cargo route.

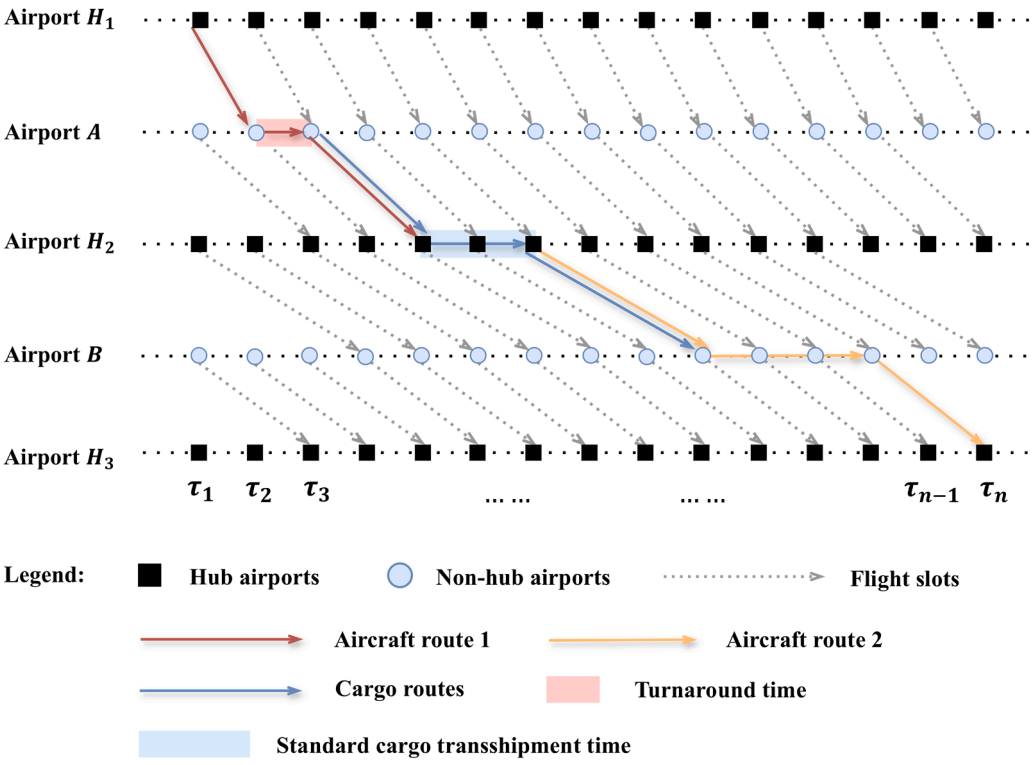

Fig. 3. Illustration of the time-space network.

Cargo aircraft operational costs consist of two primary components: (i) fixed operational costs associated with flight distance, such as aircraft depreciation, and (ii) variable operational costs related to the transportation of cargo, which include fuel surcharges and handling fees, etc. In addition to these costs, carriers also leverage the belly capacity of commercial passenger airlines to transport cargo, necessitating consideration of the booking costs associated with belly capacities. Furthermore, we permit the possibility that cargo may not be delivered by its latest arrival time. However, this operational flexibility incurs penalty costs proportional to the weight of unserved cargo. Finally, considering those costs, we can establish the objective of the problem as reducing overall operational costs within the service network.

4. Problem formulation

We present a mathematical model of the Air Cargo Service Network Scheduling problem described in the previous section. We denote our network as a directed network $D = (\mathcal{N}, \mathcal{A})$ where each node $n \in \mathcal{N}$ is an airport, and each arc $(o, d) \in \mathcal{A}$ is a point-to-point move, with $o \in \mathcal{N}$ and $d \in \mathcal{N}$. To model the temporal attributes inherent to our problem, we presume a discretization of time $T = \{\tau_1, \tau_2, \ldots, \tau_m\}$ that includes m homogeneous time periods with duration $\Delta\tau$, and spans a timeline $\left[\underline{T}, \overline{T}\right]$, during which cargo aircraft are allowed to operate. Let $\mathcal{N}_T$ denote the node set in the time-space network, which consists of nodes of the form $(i, \tau_j), i \in \mathcal{N}, \tau_j \in T$. Specifically, the node set $\mathcal{N}_T$ is obtained by duplicating each airport $i \in \mathcal{N}$ at each discrete time $\tau_j \in T$. Afterwards, for each airport $o \in \mathcal{N}$, each discrete time $\tau_i \in T$, and each arc $(o, d) \in \mathcal{A}$, we generate a flight slot l from o at time τ_i to d at time $\tau_j \in T$, with τ_j being the earliest time point of T larger than τ_i plus the travel time from o to d. Then, we can get the set of flight slots, referred to as L. Each flight slot $l \in L$ is characterized by (i) an origin airport $o^l \in \mathcal{N}$, (ii) a destination airport $d^l \in \mathcal{N}$, (iii) a departure time $st^l \in T$, and (iv) an arrival time $et^l \in T$. Thus, we can extend the aforementioned network to a time-space version $\mathcal{D}_T = (\mathcal{N}_T, L)$. Fig. 3 depicts the time-space network, where dashed arrows indicate cargo flight slots. The red line arrows and the orange line arrows represent aircraft routes, while the blue line arrows illustrate cargo routes utilizing these flight slots. Next, we provide detailed descriptions of the generation process for both aircraft and cargo routes.

4.1. Aircraft routes

We consider that the fleet includes cargo aircraft of different types, denoting the set of aircraft types as K. For each aircraft type $k \in K$, there is N_k aircraft of that type with a capacity v_k in the fleet. A cargo aircraft route is defined as a set of consecutive flight slots. For a cargo aircraft route to be valid, the origin airport of its first flight and the destination airport of its last flight must be hubs. We let P denote the set of feasible aircraft routes. We next define the requirement that aircraft routes must observe.

Let the parameter a denote the maximum number of flight slots that an aircraft route can encompass within the allowable operational duration, denoted as $T_{\max}$. This limit a ensures that the cumulative sum of the travel times of all flight slots in the route of any aircraft, plus the turnaround time between them, does not exceed $T_{\max}$. Consequently, an aircraft route p can be expressed as a sequence of flight slots $\{l_1, l_2, \ldots, l_j\}, \exists j \leq a$, that satisfy the requisite time and space constraints. This means that, for each pair of consecutive flight slots, the arrival airport of the first flight slot needs to be the departure airport of the second flight slot. In addition, a minimum transshipment time of t_{trans} must be observed between the arrival time of a flight slot and the departure time of the following flight slot. We assume this minimum transshipment time (i.e., turnaround time) t_{trans} is uniform across all aircraft types.

Considering an aircraft route p, let l_m and l_n be two consecutive flight slots in this route's sequence. Let (st^{l_m}, et^{l_m}) and (st^{l_n}, et^{l_n}) be their respective departure and arrival times. Formally, these adjacent flight slots must satisfy the following requirements:

$$o^{l_n} = d^{l_m} \tag{1a}$$

$$st^{l_n} - et^{l_m} \geq t_{trans}. \tag{1b}$$

Similar to the work of Yıldız and Savelsbergh (2022), each aircraft route p is parameterized by aircraft type k, meaning identical topological sequences of flight slots may constitute distinct aircraft routes when associated with different aircraft type. Thus, we further define $P_k^l, l \in L, k \in K$ as the set of feasible aircraft routes that pass through the flight slot l operated by a type-k aircraft. Fig. 3 illustrates two aircraft routes, represented by the red and orange arrows: one from airport H_1 to airport H_2, and another from airport H_2 to airport H_3. Furthermore, for cargo demands that have to be transported between these (O,D) pairs, these aircraft routes can be also utilized as cargo routes through the implementation of through cargo connections. Specifically, aircraft route 1 enables a through cargo connection for the cargo demand from airport H_1 to airport H_2, while aircraft route 2 enables a through cargo connection for the cargo demand from airport H_2 to airport H_3. Besides, there exists a cargo route for the transportation of cargo demands between airport A and airport B, which includes a standard transshipment connection, as illustrated by the blue arrows in Fig. 3.

4.2. Cargo routes

We are given a set of cargo demands Q. For each $q \in Q$, we assume that the corresponding ready time st^q, latest delivery time et^q, weight w^q, origin airport o^q and destination airport d^q are known. Failing to deliver a cargo before its latest delivery time yields a penalty per unit of weight η^q. Generating cargo routes via cargo aircraft is analogous to generating aircraft routes, as both procedures involve the identification of consecutive flight slots that meet spatial and time requirements while not exceeding the maximum number of segments. This problem is set within a nighttime operational window. In practice, the latest pick-up deadlines for express freight at consolidation centers and the earliest delivery service times tend to fall outside this time window. Given this operational characteristic, we simplify the demand structure for modeling purposes by assuming that only one cargo demand exists for each (O, D) pair. That is, the set of demands for a given (O, D) pair is aggregated into a single cargo q.

A cargo route, denoted by r, is defined as a sequence of flight operations. Each operation is a pair consisting of a flight slot and its assigned aircraft type. Formally, we represent a cargo route as: $r = \{(l_1, k_1), (l_2, k_2), \ldots, (l_b, k_b)\}$, where l_i is a flight slot in the sequence, and $k_i \in K$ is the aircraft type assigned to it. Note that the aircraft types $k_1, k_2, \ldots, k_b$ for different flight slots in the sequence can be either the same or different. Let (o^{l_1}, d^{l_1}), (st^{l_1}, et^{l_1}), (o^{l_b}, d^{l_b}), (st^{l_b}, et^{l_b}) be the airport of origin, destination, departure time, and arrival time of the flight slots l_1 and l_b, respectively. Subsequently, the following requirements must be met in addition to requirements (1a) and (1b).

$$o^{l_1} = o^q \tag{2a}$$

$$d^{l_b} = d^q \tag{2b}$$

$$st^q \leq st^{l_1} \tag{2c}$$

$$et^q \geq et^{l_b}. \tag{2d}$$

In details, this generation process begins with the construction of an initial pool of candidate cargo routes, formed by the sequential linking of individual cargo flight slots with specific aircraft types. The formation of these candidates is rigorously governed by the principles: (i) spatial continuity is maintained by ensuring consecutive flight slots connect at the same airport and temporal feasibility is guaranteed by respecting the minimum aircraft turnaround times between flight slots, and (ii) overarching compliance is achieved by ensuring the route's entire duration is contained within the cargo's specific shipment window.

Following the generation of this spatiotemporally feasible candidate set, each cargo route undergoes a validation phase to determine its operational viability. A candidate cargo route is formally deemed valid only if all of its constituent connections between sequential flight slots can be definitively classified. Specifically, every connection within a cargo route must qualify as either a through cargo connection or a standard cargo connection, which we formally define in the previous sections. The detailed cargo route generation algorithm is shown in Appendix A.

We let R denote the set of cargo routes involving cargo aircraft. Let $r \in R$ represent a cargo route composed of flight slots. Transporting a cargo $q \in Q$ along a route r through dedicated aircraft yields a per unit of weight cost u_r^q. Since a cargo route r consists of a sequence of flight slots, where each flight slot $l \in L$ can be associated with a specific aircraft type $k \in K$, we define R_k^l as the set of cargo routes where the cargo flight l can be performed and operated by an aircraft of type k. We further define Q_k^l as the set of cargo demands that can be transported via cargo flight l operated by the type-k aircraft.

We define the set of through cargo connections, TC, as a set of 4-tuples (r, l_m, l_n, k). An element $(r, l_m, l_n, k) \in TC$ exists if and only if cargo route r contains a consecutive connection from l_m to l_n, and both of these flight slots are operated by the same aircraft type k. Formally, the set TC is defined as:

$$TC = \left\{(r, l_m, l_n, k) \mid r \in R, k \in K, (l_m, k) \text{ and } (l_n, k) \text{ are consecutive pairs in } r\right\}.$$

Besides, (l_m, l_n) can be consecutive pairs in an aircraft route p. Thus, for each through connection pair (l_m, l_n) operated by the type-k aircraft, we can also identify $P^k_{(l_m,l_n)} \subseteq P$ which represents the set of aircraft routes operated by the type-k aircraft that also incorporate the through connection pair (l_m, l_n).

Regarding belly capacity, let B represent the set of belly capacity resources. Transporting a cargo $q \in Q$ through the resource $b \in B$ induces a per unit of weight cost β_b. We assume β_b depends solely on the specific belly capacity b rather than the specific cargo q, as this cost is determined by the service contract with commercial airlines. For each $q \in Q$, we let $B_q \subseteq B$ denote the subset of belly capacity resources eligible for cargo q. For each $b \in B$, we let Q_b represent the set of cargo $q \in Q$ that can be transported by belly capacity resource b. For each $b \in B$, we let ϖ_b refer to the associated capacity.

4.3. Mathematical model

Table 2 provides a detailed overview of the symbols and notation used in our model. We define the binary variable $f^l_k, k \in K, l \in L$ such that $f^l_k = 1$ if the flight slot l operated by a type-k aircraft is selected, and $f^l_k = 0$ otherwise. We let the binary variable $x^p_k, p \in P, k \in K$ indicate whether or not the aircraft route p is operated by a type-k aircraft. We let the integer variable $y^q_r \in \mathbb{N}, q \in Q, r \in R$ denote the weight of cargo q that follows the route r. We let the integer variable $\gamma^q_{(r,l_m,l_n,k)} \in \mathbb{N}, q \in Q, (r, l_m, l_n, k) \in TC$, denote the weight of cargo q that follows the through cargo connection (r, l_m, l_n, k). We let the integer variable $\varphi^q_b \in \mathbb{N}, q \in Q, b \in B$ denote the weight of cargo q that is transported by the belly capacity b. We let the integer variable $z^q \in \mathbb{N}, q \in Q$ denote the weight of cargo q that is not delivered.

The problem can be formulated as:

$$\min \quad \sum_{q\in Q}\sum_{r\in R} u^q_r y^q_r + \sum_{q\in Q}\sum_{b\in B} \beta_b \varphi^q_b + \sum_{q\in Q} \eta^q z^q + \sum_{k\in K}\sum_{l\in L} c^l_k f^l_k \tag{3a}$$

$$s.t. \sum_{k\in K} f^l_k \le 1, \qquad \forall l \in L \tag{3b}$$

$$\sum_{p\in P} x^p_k \le N_k, \qquad \forall k \in K \tag{3c}$$

$$\sum_{o^p=i, p\in P} x^p_k - \sum_{d^p=i, p\in P} x^p_k = 0, \qquad \forall i \in H, k \in K \tag{3d}$$

$$\sum_{p\in P^l_k} x^p_k = f^l_k, \qquad \forall l \in L, k \in K \tag{3e}$$

$$\sum_{r\in R} y^q_r + \sum_{b\in B_q} \varphi^q_b + z^q = w^q, \qquad \forall q \in Q \tag{3f}$$

$$\sum_{q\in Q}\sum_{r\in R^l_k} y^q_r \le v^k f^l_k, \qquad \forall l \in L, k \in K \tag{3g}$$

$$\sum_{q\in Q_b} \varphi^q_b \le \varpi_b, \qquad \forall b \in B \tag{3h}$$

$$\gamma^q_{(r,l_m,l_n,k)} = y^q_r, \qquad \forall (r, l_m, l_n, k) \in TC, q \in Q^{l_m}_k \cup Q^{l_n}_k \tag{3i}$$

$$\gamma^q_{(r,l_m,l_n,k)} \le \min\{w^q, v_k\} \sum_{p\in P^k_{(l_m,l_n)}} x^p_k, \qquad \forall (r, l_m, l_n, k) \in TC, q \in Q^{l_m}_k \cup Q^{l_n}_k \tag{3j}$$

$$x^p_k \in \{0, 1\}, \qquad \forall p \in P, k \in K \tag{3k}$$

$$f^l_k \in \{0, 1\}, \qquad \forall l \in L, k \in K \tag{3l}$$

$$\gamma^q_{(r,l_m,l_n,k)} \in \mathbb{N}, \qquad \forall (r, l_m, l_n, k) \in TC, q \in Q^{l_m}_k \cup Q^{l_n}_k \tag{3m}$$

$$y^q_r \in \mathbb{N}, \qquad \forall q \in Q, r \in R \tag{3n}$$

$$\varphi^q_b \in \mathbb{N}, \qquad \forall q \in Q, b \in B \tag{3o}$$

$$z^q \in \mathbb{N}, \qquad \forall q \in Q. \tag{3p}$$

The objective function (3a) aims to minimize the total cost, which equals the sum of: (i) cargo transportation costs associated with dedicated aircraft, (ii) cargo transportation costs associated with belly resources, (iii) penalty costs, and (iv) transportation costs of dedicated aircraft.

Constraints (3b) – (3e) are the fleet scheduling constraints. Constraints (3b) indicate that each flight slot can only be operated by one type of aircraft at most. Constraints (3c) guarantee that the number of type-k cargo aircraft used cannot exceed the number of

Table 2
Notations.

Notations	Description
Sets and indices	
L	Set of all flight slots
l	Index of flight slot
K	Set of aircraft types
k	Index of aircraft type
H	Set of hub airports
i	Index of hub airport
P	Set of all feasible aircraft routes
p	Index of feasible aircraft route
P_k^l	Set of feasible aircraft routes through cargo flight slot l operated by the type-k aircraft
Q	Set of all cargo demand
q	Index of cargo demand
R	Set of feasible cargo routes through cargo flight slots
r	Index of feasible cargo route through cargo flight slots
R_k^l	Set of feasible cargo routes through cargo flight slot l operated by the type-k aircraft
Q_k^l	Set of cargo demand that can be transported via cargo flight slot l operated by the type-k aircraft
TC	Set of through cargo connections
(r, l_m, l_n, k)	Index of through cargo connection that represent a through cargo connection pair $((l_m, k), (l_n, k))$ in a cargo route r
$P^k_{(l_m,l_n)}$	Set of feasible aircraft routes operated by type-k aircraft involving through connection pair (l_m, l_n)
B	Set of all belly capacity resources of passenger airlines
b	Index of belly capacity resource
B_q	Set of all belly capacity resources of passenger airlines that can transport cargo demand q
Q_b	Set of cargo demand that can be transported via the belly capacity resource b
Parameters	
ϖ_b	Capacity of belly capacity resource b
v_k	Capacity of the type-k aircraft
c_k^l	The fixed cost of operating flight slot l with the type-k aircraft
N_k	The number of type-k aircraft
w^q	Weight of cargo demand q
u_r^q	Unit cost of cargo demand q transported using route r through dedicated aircraft
η^q	Unit penalty for weight not carried of cargo demand q
β_b	Unit cost of cargo transported by the belly capacity b
o^p	The origin airport of aircraft route p
d^p	The destination airport of aircraft route p
Decision variables	
x_k^p	$\in \{0, 1\}$, 1 if aircraft route p is operated by the type-k aircraft, and 0 otherwise
f_k^l	$\in \{0, 1\}$, 1 if flight slot l is selected and served by the type-k aircraft, and 0 otherwise
y_r^q	$\in \mathbb{N}$, Weight of cargo demand q transported by cargo route r
$\gamma^q_{(r,l_m,l_n,k)}$	$\in \mathbb{N}$, Weight of cargo demand q transported by the through cargo connection (r, l_m, l_n, k)
φ_b^q	$\in \mathbb{N}$, Weight of cargo demand q carried by belly capacity resource b
z^q	$\in \mathbb{N}$, Weight of cargo demand q that cannot be delivered.

type-k cargo aircraft available. Constraints (3d) represent the flow conservation constraints for each aircraft type, indicating that the number of aircraft with a specific type departing from a hub airport must equal the number of aircraft with that type arriving at that airport. Constraints (3e) link the aircraft route variables and the flight slot variables. They guarantee that a flight slot is operated by a type-k aircraft if and only if a route composed with that flight slot is assigned to a type-k aircraft.

Constraints (3f) – (3h) are the cargo routing constraints. Constraints (3f) guarantee that, for each cargo, the sum of the weight served through the cargo aircraft, the weight served through belly resource, and the unserved weight, equals the total weight. Constraints (3g) enforce capacity requirements for the dedicated aircraft. Constraints (3h) enforce capacity requirements for the belly resources.

Constraints (3i) – (3j) are the through cargo connection constraints. Constraints (3i) require that the flow of cargo q on the through cargo connection (r, l_m, l_n, k) equals the total weight of cargo transported on route r. Furthermore, constraints (3j) ensure that such through cargo connection flow can only occur if an aircraft route $p \in P^k_{(l_m,l_n)}$ of aircraft type k is selected to operate the connection (l_m, l_n).

4.4. Model analysis

In this section, we provide a theorem regarding the totally unimodular property of a restricted version of our problem, which we use in our solution algorithm to reduce the computational burden. We observe the unimodular property within our model when certain variables are fixed, as detailed in the following theorem.

Theorem 1. *Assuming that the values are set for all the binary variables x_k^p and f_k^l, the constraint coefficient matrix associated with model (3) is a totally unimodular matrix (TUM).*

Proof. Through an analysis of the model's structure with specific binary variables assigned appropriate values, it is determined that the constraint coefficient matrix is a TUM, based on the Ghouila-Houri characterization (Ghouila-Houri 1964). Recall that, a matrix is classified as TUM if, for any collection of its columns (rows), those columns (rows) can be divided into two subsets such that the difference between the sums of the columns (rows) in each subset results in a vector with entries limited to 0, +1, or −1. We begin by analyzing each constraint individually, focusing on constraints (3f) – (3j), as constraints (3b) – (3e) are constant constraints when given fixed values of x_k^p and f_k^l. For each of these constraints, we identify the corresponding constraint coefficient matrix. By evaluating the structure of these matrices, we assess whether they permit the partitioning of columns (or rows) as required by the Ghouila-Houri characterization. If all constraints conform to this condition, we can conclude that the constraint coefficient matrices preserve the TUM property (Molenbruch et al. 2023). Further details of the proof can be found in Appendix B. □

According to this theorem, certain integer variables can be relaxed to continuous variables without compromising the quality of the solutions under specific conditions.

5. Column-generation based methodology

We propose a solution approach based on Column Generation (CG). In this section, we first introduce a CG strategy for solving the linear programming (LP) relaxation of the model to obtain a subset of aircraft routes. Subsequently, to obtain a high-quality integer solution, we first apply recombination and duplication strategies to expand the pool of aircraft routes. Based on this expanded set, we then generate the corresponding compatible cargo routes. Finally, leveraging the TUM properties established in Theorem 1, we formulate a restricted mixed-integer program (MIP) by restoring integrality constraints only on the binary variables to obtain the final integer solution. We next describe the proposed heuristic in detail.

5.1. Column generation

The primary advantage of CG lies in its effectiveness in addressing linear optimization problems with an exponential number of variables. The CG algorithm starts by solving a version of the original problem involving a subset of the variables, referred to as the Restricted Master Problem (RMP). Subsequently, the algorithm solves a pricing problem to identify new variables (or columns) that have the potential to enhance the objective function of the RMP. The process repeats until no improving variables can be identified and terminates with an optimal solution to the linear program.

5.1.1. Restricted master problem

To enhance computational efficiency, we first construct our initial RMP by including a constraint that tightens the potential values of variables y_r^q, which represent the flow of cargo $q \in Q$ on a candidate flight route $r \in R$. Then, we construct an initial set of cargo flight routes according to the tighter values of variables y_r^q, denoted as $\underline{R} \subseteq R$. This tightening rule is based on both physical limitations and economic dominance.

First, the physical capacity of any cargo route r, denoted by V_r, is limited by the bottleneck of its assigned aircraft fleet:

$$V_r = \min_{k \in K_r} \{v_k\},$$

where K_r is the set of aircraft types used for cargo route r. Second, we apply an economic routing logic based on our objective of minimizing total operational cost. For any given cargo q, if the variable cost u_r^q of using a dedicated cargo flight route r exceeds the benchmark price β^b of alternative belly capacity ($b \in B_q$ being the set of belly capacity available to q), the cost-minimization principle dictates that this cargo will preferentially utilize the cheaper belly capacity first. Consequently, the flow y_r^q on the more expensive cargo route r must be bounded by the demand shortfall. This shortfall is the residual demand w^q that cannot be met by the total available economical belly capacity $U^B(q)$, where $U^B(q) = \sum_{b \in B_q : u_r^q > \beta^b} \varpi^b$. This economic upper bound is formulated as $\max(0, w^q - U^B(q))$.

By integrating these limits, we establish a tighter composite upper bound $\bar{y}_r^q$ for each variable in the initial RMP:

$$\bar{y}_r^q = \min\left(\max(0, w^q - U^B(q)), V_r\right).$$

A route $r \in R$ is eliminated if it proves non-viable, meaning it cannot carry any cargo after applying these rules (i.e., $\bar{y}_r^q = 0$). The initial route set $\underline{R}$ is thus defined as the set of all valid routes:

$$\underline{R} = \left\{ r \in R \mid \sum_{q \in Q} \bar{y}_r^q > 0 \right\}.$$

The initial RMP is then constructed using only the routes $r \in \underline{R}$. For all variables y_r^q included in this RMP ($r \in \underline{R}, q \in Q$), their respective upper bound constraints are tightened by adding the following constraints:

$$y_r^q \le \bar{y}_r^q, \qquad \forall q \in Q, r \in \underline{R}. \tag{4}$$

Correspondingly, the set R_k^l is also limited to its subset $\underline{R}_k^l$, defined as $\underline{R}_k^l = \{r \in R_k^l \mid r \in \underline{R}\}$. Similarly, the set of through cargo connection TC must be restricted to $\underline{TC}$, which includes only those connections associated with the cargo routes currently in the initial RMP as $\underline{TC} = \{(r, l_m, l_n, k) \in TC \mid r \in \underline{R}\}$.

Let P_s be the restricted subset of aircraft routes, the RMP is formulated as follows.

$$\min \quad \sum_{q\in Q}\sum_{r\in \underline{R}} u_r^q y_r^q + \sum_{q\in Q}\sum_{b\in B} \beta_b \varphi_b^q + \sum_{q\in Q} \eta^q z^q + \sum_{k\in K}\sum_{l\in L} c_k^l f_k^l \tag{5a}$$

$s.t.$

$(3b), (3h), (4)$

$$\sum_{p\in P_s} x_k^p \le N_k, \qquad \forall k \in K \tag{5b}$$

$$\sum_{o^p=i, p\in P_s} x_k^p - \sum_{d^p=i, p\in P_s} x_k^p = 0, \qquad \forall i \in H, k \in K \tag{5c}$$

$$\sum_{p\in P_k^l \cap P_s} x_k^p = f_k^l, \qquad \forall l \in L, k \in K \tag{5d}$$

$$\sum_{r\in \underline{R}} y_r^q + \sum_{b\in B_q} \varphi_b^q + z^q = w^q, \qquad \forall q \in Q \tag{5e}$$

$$\sum_{q\in Q}\sum_{r\in \underline{R}_k^l} y_r^q \le v^k f_k^l, \qquad \forall l \in L, k \in K \tag{5f}$$

$$\gamma^q_{(r,l_m,l_n,k)} = y_r^q, \qquad \forall (r,l_m,l_n,k) \in \underline{TC}, q \in Q_k^{l_m} \cup Q_k^{l_n} \tag{5g}$$

$$\gamma^q_{(r,l_m,l_n,k)} \le \min\{w^q, v_k\} \sum_{p\in P^k_{(l_m,l_n)} \cap P_s} x_k^p, \qquad \forall (r,l_m,l_n,k) \in \underline{TC}, q \in Q_k^{l_m} \cup Q_k^{l_n} \tag{5h}$$

$$x_k^p \ge 0, \qquad \forall p \in P_s, k \in K \tag{5i}$$

$$0 \le f_k^l \le 1, \qquad \forall l \in L, k \in K \tag{5j}$$

$$y_r^q \ge 0, \qquad \forall q \in Q, r \in \underline{R} \tag{5k}$$

$$\gamma^q_{(r,l_m,l_n,k)} \ge 0, \qquad \forall (r,l_m,l_n,k) \in \underline{TC}, q \in Q_k^{l_m} \cup Q_k^{l_n} \tag{5l}$$

$$\varphi_b^q \ge 0, \qquad \forall q \in Q, b \in B \tag{5m}$$

$$z^q \ge 0, \qquad \forall q \in Q. \tag{5n}$$

5.1.2. Pricing problem

The pricing problem aims to identify new aircraft routes with the lowest reduced costs. If aircraft routes with negative reduced costs are found, they are added to the RMP. If there are no aircraft routes with negative reduced cost, the optimal solution to the linear program is found, and the algorithm terminates. Let λ_k, ρ_k^i, ψ_k^l, $\phi^q_{(r,l_m,l_n,k)}$ be the dual variables corresponding to constraints(5b) – (5d) and (5h), respectively. Then, the reduced cost δ_k^p of the aircraft route p operated by the type-k aircraft can be given as follows:

$$\delta_k^p = 0 - \Big(\sum_{l\in p} \psi_k^l + \lambda_k + \rho_k^{o^p} - \rho_k^{q^p} - e_k^p\Big), \tag{6}$$

in which, $e_k^p = 0$ if the aircraft route p operate by the type-k aircraft does not operate any through connection (l_m, l_n) that can be utilized in a cargo route r as a through cargo connection (r, l_m, l_n, k). Otherwise, $e_k^p = \sum_{(l_m,l_n)\in A(p)} \sum_{q\in Q_k^{l_m}\cup Q_k^{l_n}} \sum_{r\in R^k_{(l_m,l_n)}} w^q \phi^q_{(r,l_m,l_n,k)}$. Here, we first define $A(p)$ as the set of flight slots pair (l_m, l_n) that constitutes the path p. For example, if path p traverses flight slots $l_1 \to l_2 \to l_3$, then $A(p) = \{(l_1, l_2), (l_2, l_3)\}$. Next, for the given aircraft type k and any given flight slots pair (i.e., through connection) $(l_m, l_n) \in A(p)$, we define $R^k_{(l_m,l_n)}$ as the set contains all associated cargo route r from TC, such that: $R^k_{(l_m,l_n)} = \{r \mid (r, l_m, l_n, k) \in TC\}$.

5.1.3. Solving the pricing problem

In this section, we demonstrate how the pricing problem can be transformed into a longest path finding problem. For a specific type-k cargo aircraft, we can define a directed acyclic graph (DAG) $G_k(S, A)$ as follows. S represents the set of nodes in this graph. It includes a dummy source node $\underline{\pi}$, a dummy sink node $\bar{\pi}$, as well as a node for each flight slot $l \in L_k$, where $L_k \subseteq L$ denotes the set of these flight slots operated by a type-k aircraft. A denotes the set of arcs in this graph, which can be divided into three distinct sets: $A_{\underline{\pi}}$, A_{L_k}, and $A_{\bar{\pi}}$, as follows.

- $A_{\underline{\pi}} = \{(\underline{\pi}, l) : l \in L_k, o^l \in H\}$
- $A_{L_k} = \{(l, \bar{l}) : l, \bar{l} \in L_k$ and satisfy constraints (1a) and (1b)$\}$
- $A_{\bar{\pi}} = \{(l, \bar{\pi}) : l \in L_k, d^l \in H\}$

The set $A_{\underline{\pi}}$ comprises the start arcs that connect the dummy source node to the flight slots originating from hub airports. The set A_{L_k} includes the connected arcs, encompassing the flight slots that satisfy the specified temporal and spatial constraints. The set $A_{\bar{\pi}}$ consists of the end arcs, which link the flight slots arriving at hub airports to the dummy sink node. Furthermore, weights are assigned to each of the defined sets of arcs, as detailed below.

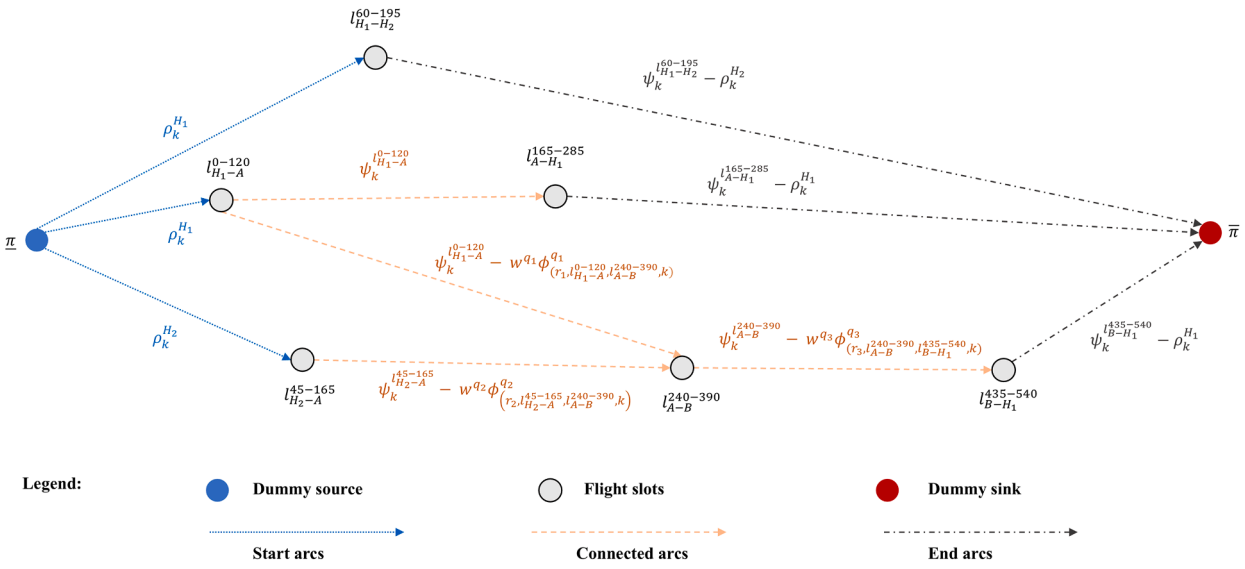

Fig. 4. Illustration of a simple DAG with a specific aircraft type.

- $W_k^{(\underline{\pi},l)} = \rho_k^{o^l}$
- $W_k^{(l,\bar{l})} = \psi_k^l - e_k^{(l,\bar{l})}$
- $W_k^{(l,\bar{\pi})} = \psi_k^l - \rho_k^{d^l}$.

If $(l,\bar{l})$ is a through connection operated by the type-k aircraft, then $e_k^{(l,\bar{l})} = \sum_{q\in Q_k^l \cup Q_k^{\bar{l}}} \sum_{r\in R_{(l,\bar{l})}^k} u^q \phi_{(r,l,\bar{l},k)}^q$; otherwise, $e_k^{(l,\bar{l})} = 0$. Thus, we can get $e_k^p = \sum_{(l,\bar{l})\in A(p)} e_k^{(l,\bar{l})}$. Based on the graph representation provided and given a type k, the total cost of a path p from the dummy source node to the dummy sink node is obtained by summing the weights associated with the individual arcs (i.e., through connection) along that path:

$$\eta_k^p = \sum_{l\in p} \psi_k^l + \rho_k^{o^p} - \rho_k^{d^p} - e_k^p. \tag{7}$$

Then, we can observe the following relationship between η_k^p and δ_k^p, the reduced cost of aircraft route p operated by the specific type-k aircraft, namely: $\delta_k^p = 0 - (\eta_k^p + \lambda_k)$. Therefore, the pricing problem of finding the cargo aircraft path corresponding to the most negative reduced cost can be transformed into the problem of identifying the path with the highest weight in the associated DAG $G_k(S,A)$ with the specific aircraft type k.

Fig. 4 illustrates an example of a DAG specific to aircraft type k, within a simplified service network comprising two hub airports, H_1 and H_2, and two non-hub airports, A and B. In this context, let l_{o-d}^{st-et} denote a generated flight slot, where st represents the departure time, et indicates the arrival time, o denotes the origin airport, and d signifies the destination airport. For instance, $l_{H_1-A}^{0-120}$ represents a flight slot that departs from airport H_1 at time 0 (indicating the beginning of the planning period) and arrives at airport A at time 120 (120 minutes after the planning period starts). There are clearly four feasible routes from the dummy source $\underline{\pi}$ to the sink $\bar{\pi}$. Considering there are three cargo demands, labeled q_1, q_2, and q_3, with the corresponding origin-destination pairs being H_1 to B, H_2 to B, and A to H_1, respectively. Then, for each demand, we can generate a corresponding feasible through cargo connection, designated as $(r_1, l_{H_1-A}^{0-120}, l_{A-B}^{240-390}, k)$, $(r_2, l_{H_2-A}^{45-165}, l_{A-B}^{240-390}, k)$, and $(r_3, l_{A-B}^{240-390}, l_{B-H_1}^{435-540}, k)$, respectively. Utilizing the aforementioned approach, we can determine the weight of each arc within this DAG, as depicted in the Fig. 4.

To accelerate the iteration process, our objective is to identify multiple aircraft routes within a DAG that maximize the associated cost. To achieve this, we propose a Δ-longest path algorithm to find multiple paths in each iteration, which is based on the DAG longest path algorithm and dynamic programming (DP) (Bellman 1962, Christofides 1975, Bang-Jensen and Gutin 2008). Here, Δ denotes the number of paths to be identified in each iteration. For each $s \in S$ and $i \in \{1,2,\ldots,\Delta\}$, we define $path[s][i]$ and $dis[s][i]$ to represent the i_{th} longest path from dummy source to node s and its corresponding cost, respectively. For each arc $(s,n) \in A$, $s,n \in S$, let $c_{(s,n)}$ denote the cost associated with arc (s,n). The dynamic programming recurrence relation for $dis[n][i]$ can be expressed as follows:

$$dis[n][i] = \max_{\substack{(s,n)\in A,\\ j\in\{1,2,\ldots,i\}}} \{dis[s][j] + c_{(s,n)} \mid path[s][j] + [(s,n)] \neq path[n][k], \forall k \in \{1,2,\ldots,i-1\}\}.$$

The detailed approach is outlined in Algorithm 1. At each iteration of the CG algorithm, and for each aircraft type k, we construct the corresponding DAG $G_k(S, A)$ and apply this algorithm to identify the Δ longest paths.

Algorithm 1 Δ-longest path algorithm.

Input: DAG $G_k(S, A)$, the value of Δ, set of restricted aircraft paths P_s
Output: Δ paths not in P_s
State Initialization.
$\Delta' \leftarrow \Delta + |P_s|$
$dis[\pi][1] \leftarrow 0$; $dis[\pi][i] \leftarrow -\infty, \forall i \in \{2, \dots, \Delta'\}$;
$dis[s][i] \leftarrow -\infty, \forall s \in S \setminus \{\pi\}, i \in \{1, \dots, \Delta'\}$
$path[s][i] \leftarrow [\,], \forall s \in S, i \in \{1, \dots, \Delta'\}$
DP-based Iteration.
for $i = 1$ **to** Δ' **do**
  for each node $s \in S$ **do**
    for each arc $(s, n) \in A$ **do**
      for $j = 1$ **to** i **do**
        if $dis[n][i] < dis[s][j] + c_{(s,n)}$ **and** $\{path[s][j] + [(s, n)] \neq path[n][k], \forall k \in \{1, 2, \dots, i-1\}\}$ **then**
          $dis[n][i] \leftarrow dis[s][j] + c_{(s,n)}$
          $path[n][i] \leftarrow path[s][j] + [(s, n)]$
According to dis and $path$, collect top Δ' paths, filter out paths in P_s, and output Δ paths.

5.2. CG based heuristic

We next describe how our solution approach operates. It first applies the CG algorithm. Specifically, it solves the RMP at each iteration and determines the corresponding dual variable values. The DAG is generated for each aircraft type k, and the Δ longest paths are identified. Subsequently, amongst the identified paths, all those with negative reduced costs are added to the RMP and the aforementioned process is repeated. If none of the paths found yield a negative reduced cost, the LP solution is obtained. At this stage, we construct a final MIP to determine an integer-feasible solution to our problem. The set of cargo aircraft routes generated by the CG iterative process serves as the initial pool of available cargo aircraft routes. To enhance this pool and explore a more promising solution space, we employ recombination and duplication strategies on these CG-generated cargo aircraft routes. We then reintroduce the original integrality constraints (i.e., on variables $\mathbf{x}$ and $\mathbf{f}$) into the restricted master problem. This final MIP is restricted to using only the aircraft routes from this expanded set and the corresponding cargo routes derived from them. The process is detailed in the following sections.

5.2.1. Aircraft routes regeneration

Based on the aircraft routes obtained from the column generation algorithm, we can identify flight slots appear in high-quality solutions. Thus, given the slots that appear in aircraft routes generated through column generation, we generate new aircraft routes and incorporate them into the set of aircraft routes considered by the MIP. For instance, let us assume that two are derived from the CG iteration, namely, $l_1 - l_2 - l_3$ and $l_4 - l_5 - l_6$. If l_1 and l_6 can be connected, i.e., l_1 and l_6 satisfy the spatial-temporal constraints (1a) and (1b), a new aircraft route $l_1 - l_6$ is generated and included in the MIP.

5.2.2. Aircraft routes duplication by aircraft type

Furthermore, we implement a type-duplication approach to expand the pool of aircraft routes available to the MIP. This approach is applied to an existing aircraft route $p = \{l_1, l_2, l_3\}$, which is operated by an aircraft of initial aircraft type $k' \in K$. We then iterate through the set of aircraft types. For each alternative type $k \in K \setminus \{k'\}$, we generate a new route candidate. This new candidate uses the identical flight slots p but assigns it to type k. This new route-type pairing is then represented by a variable x_k^p, which is included in the MIP candidate set, provided it is not already present.

5.2.3. Generation of the final cargo route set

The construction of the final MIP requires a corresponding set of cargo routes using cargo flights. Based on the comprehensive set of cargo aircraft routes assembled in the previous steps, we can get all flight slots and corresponding assigned aircraft type used by those aircraft routes. Then, we explicitly generate the set of feasible cargo routes, denoted by R_{CG}, using those flight slots and aircraft types. This generation is performed using the specific Cargo Route Generation Algorithm detailed in the Appendix A. The resulting set R_{CG} is a subset of $\underline{R}$ (i.e., $R_{CG} \subseteq \underline{R}$). This restricted set R_{CG} is then used exclusively to define the cargo flow variables and associated constraints within the final MIP model. In summary, the Column Generation algorithm in this study is shown as Algorithm 2.

Algorithm 2 Column Generation algorithm.

Input: The value of Δ and the empty set of aircraft route P_s to the initial RMP.
Output: The integer solution $(\boldsymbol{x}^*, \boldsymbol{f}^*, \boldsymbol{y}^*, \boldsymbol{\gamma}^*, \boldsymbol{\varphi}^*, \boldsymbol{z}^*)$ and the objective value.
Iteration.
while True **do**
    Solve RMP and get values of the corresponding dual variables.
    for each aircraft type $k \in K$ **do**
        Generate $G_k(S, A)$.
        Apply Δ-longest path algorithm to get Δ paths with their cost η_k^p.
        $\delta_k^p \leftarrow 0 - (\eta_k^p + \lambda_k)$.
    Get the aircraft routes corresponding to $\delta_k^p < 0$, add them to set P_s, and update the RMP.
    Get the lowest reduced cost $\delta_{\min}$ among all reduced cost δ_k^p.
    if $\delta_{\min} \geq 0$ **then**
        return P_s.
Final MIP.
Generate an expanded aircraft route set $\tilde{P}$ from P_s (via recombination/duplication) and define integer variables x_k^p.
Define integer variables f_k^l from all flight slots and assigned aircraft type pair (l, k) in $\tilde{P}$.
Formulate the final MIP by imposing integrality on $\boldsymbol{x}, \boldsymbol{f}$ and coupling them with the re-generated cargo flight route variables $(\boldsymbol{y}, \boldsymbol{\gamma})$ using the RMP structure.
Solve MIP and return the obtained solution.

6. Computational experiments

To validate the proposed model and algorithm, we derive instances based on data from a major air freight company in China. The operational planning period for our air cargo service network scheduling problem was established as one night, with the cargo flight time window restricted to the allowable hours for cargo aircraft operations in China. Accordingly, the temporal discretization step $\Delta\tau$ is set to 15 minutes. The time window for all cargo involves an available time of 21:00 and a due time of 8:00. In most of the literature on aircraft routing, the turnaround time is in a range of 30 to 60 minutes. Thus, we set the turnaround time t_{trans} to 45 minutes. Furthermore, for each cargo, there can be no more than two transshipments along its route and a transshipment operation between different aircraft can only be performed if the transfer time is at least of two hours (Lee et al. 2019, Zheng et al. 2023).

The first experiment involves various instances of different scales for network size to validate our proposed algorithm. The second experiment investigates the influence of through cargo connections on the models under consideration. Finally, the third experiment assesses the effect of unserved penalties within the study framework. These experiments are conducted using C++ to interface with CPLEX 12.10, running on a 2.30 GHz Intel Core Ultra 9 185H CPU with 32 GB of RAM under the Windows 11 operating system.

6.1. Instances and experimental settings

We utilized a dataset provided by our industrial partner company, encompassing daily cargo flow, available belly capacity, and corresponding airport data. This data undergo a two-stage preprocessing phase to ensure network feasibility and scope. First, we exclude any cargo demand ((O,D) pair) for which an available cargo route, operated by cargo aircraft, could not be established within a single overnight time window. Second, to account for the complexities of customs and cross-border procedures, all cross-border cargo demands are removed from consideration. The final real-size network we use as the base large-scale network is illustrated in Fig. 5. This network consists of 24 airports, which are categorized into 5 hubs and 19 non-hub airports. Additionally, in Fig. 5, the thickness of the lines is proportional to the magnitude of the cargo flow between the connected (O,D) pairs.

To evaluate the model on instances of varying scale, we generate subnetworks by sampling from the 5 primary hubs. Small-scale networks are constructed by selecting all possible combinations of 2 out of the 5 hubs, resulting in 10 distinct small-scale networks. Each of these networks includes the two selected hubs and their respectively associated spoke airports. Similarly, medium-scale networks are generated by selecting all combinations of 4 out of the 5 hubs, yielding 5 medium-scale networks. As with the large-scale network, we apply the same preprocessing filter to all subnetworks, removing any cargo demand that lacked a feasible overnight cargo route transported by cargo aircraft.

Based on these networks, we then generate a comprehensive testbed by randomly varying key operational parameters, including total cargo demands, available belly capacity, and fleet size. This process yields 50 distinct instances for each of the three scales (i.e., small, medium, and large). Table 3 provides a summary of the average characteristics for each instance scale, detailing the mean number of (O,D) pairs, fleet size, total cargo demands, and available belly capacity. In Table 3, there are three aircraft types in the fleet, namely K1 with a capacity of 14 tonnes, K2 with a capacity of 25 tonnes, and K3 with a capacity of 48 tonnes. The costs associated with the fleet and the corresponding notation settings are detailed in Tables 4 and 5.

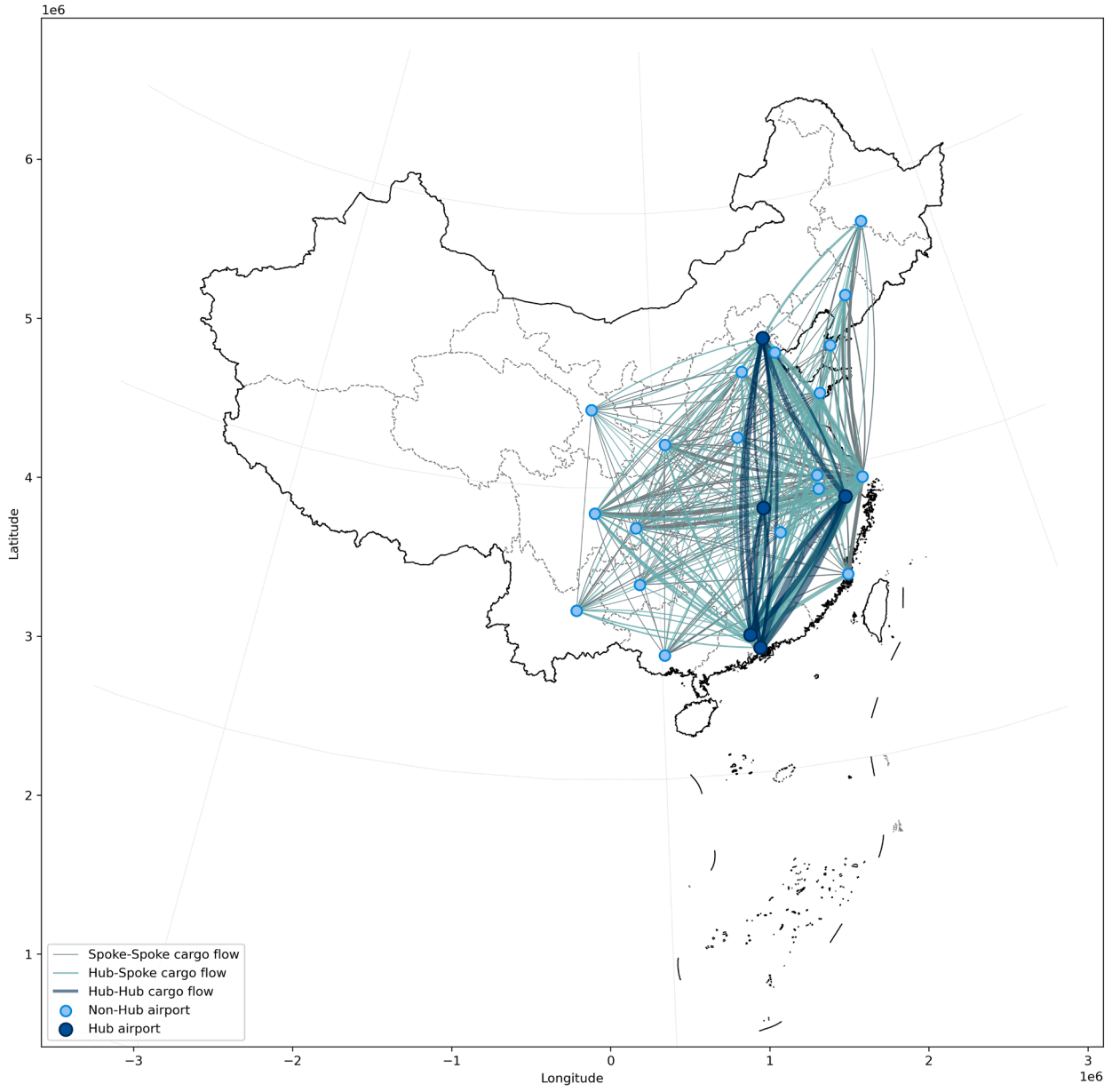

Fig. 5. Base large-scale network.

Table 3
Summary of instance characteristics by scale.

Scale	Hubs	(O, D) pairs			(O, D) demand (kg)			belly capacity (kg)			fleet size		
		Min	Max	Average	Min	Max	Average	Min	Max	Average	K1	K2	K3
Small	2	66	82	74.1	404,774	742,604	582,704.46	64,842	180,678	115,830.50	5 - 6	3 - 4	2 - 3
Medium	4	157	177	167.1	1,205,764	1,626,878	1,386,547.60	200,267	339,904	258,029.20	10 - 12	6 - 8	4 - 5
Large	5	257	257	257.0	1,890,809	2,159,095	2,042,785.80	348,882	418,562	385,217.74	18 - 23	18 - 23	5 - 8

Table 4
Fleet cost.

Notation	Description	K1	K2	K3
c_k^{fix}	Operational fixed cost	15,000	25,000	35,000
c_k^{var}	Operational variable cost (/(min × kg))	0.018	0.013	0.010
v_k	Capacity (t)	14	25	48

Table 5
Cargo and flight characteristics.

Noatation	Description	Value
β_b	Transported cost by belly (/kg)	3.5
η^q	Penalty for not transported cargo (/kg)	30.0
tra_k^l	Travel time of flight f_k^l (mins)	input data
u_r^q	Transported cost by dedicated aircraft (/kg)	$\sum_{l\in r}\sum_{k\in K}(c_k^{var}\times tra_k^l)$
c_k^l	The fixed flight operational cost	c_k^{fix}

6.2. Computational performance

This section presents the computational study designed to evaluate the performance of our proposed CG algorithm. For a fair benchmark, the CPLEX solver is applied to our MIP formulation with fully exploiting the TUM property identified in Section 4.4. We first compare the performance of our CG approach against this TUM-enhanced CPLEX model and a Benders-based method on a set of small-scale instances. This analysis is then extended to medium- and large-scale instances to assess the scalability and effectiveness of the CG algorithm compared directly with CPLEX. Finally, we investigate the specific contributions of our modeling strategies by quantifying the computational impact of leveraging the TUM property and evaluating the effectiveness of the additional tightening constraints in Section 5.1.1.

6.2.1. Experiments on small-scale instances

To evaluate our proposed CG algorithm, we first conduct a comprehensive comparison against the CPLEX solver and a Benders-based method (detailed in Appendix C) on the 50 small-scale benchmark instances. A uniform time limit of one hour (3600s) is used for all methods. For the CG algorithm, the parameter Δ within the Δ-longest path algorithm is set to 5.

Fig. 6 presents a performance comparison of our proposed CG algorithm against those two benchmarks for the small-scale instances (the detailed results are shown in Appendix D). In this figure, the x-axis indicates the different instances, and each point corresponds to the output obtained for a given method for a specific instance. Fig. 6(a) provides the computational times. The Benders-based method is omitted from this figure, as it consistently reached the 3600-second time limit on all instances. In this subfigure, *CPLEX Time* denotes the total solution time required by CPLEX, while *CG Total Time* represents the total time for the CG algorithm. The latter is composed of the CG iteration time (*CG LP Time*) and the final MIP solution time. The results indicate that our CG algorithm consistently surpasses CPLEX in computational speed. While both methods solve the simpler instances (1–34) efficiently, the performance gap becomes pronounced on the more challenging instances (35–50). For these instances, CPLEX fails to find an optimal solution, terminating at the 3600-second time limit. In contrast, our CG algorithm successfully solves 49 out of 50 instances within the time limit, leading to a significantly lower average computational time (404.34s) compared to CPLEX (1428.14s). Furthermore, the *CG LP Time* (average: 10.59s) remains small across all instances, confirming the high efficiency of the column generation process.

Fig. 6(b) analyzes the solution quality using the *IP.gap* metric. Let *Obj* denote the objective value, then, for a given method *Method*, the *IP.gap* can be calculated as: $(Method.Obj - CPLEX.Obj)/CPLEX.Obj \times 100\%$. The Benders-based method proves ineffective, yielding a prohibitively large average gap of 92.95%. Conversely, our CG algorithm demonstrates excellent performance, as its *IP.gap* remains at 0.00% for most instances and averages a negligible value of 0.03%. This result confirms that our method consistently finds solution with a quality equivalent to that of CPLEX. Additionally, for instances 38 and 44, CPLEX timed out, whereas our CG algorithm found solutions in 507.74s and 1160.90s, respectively. More importantly, the CG algorithm also achieved superior solution quality. The negative *IP.gaps* reported for our algorithm (-0.15% and -0.64%) signify that it found better integer solutions (i.e., lower objective values) than CPLEX found within the time limit.

We also note that for nine of the small-scale instances in Fig. 6, the final solution quality of the CG algorithm is inferior to that of CPLEX, even though our CG algorithm can get the solutions quickly. To facilitate a fairer comparison of the solution performance, we record the time at which the best integer solution is found during the final MIP solve of the CG algorithm, denoted as T_{int}. Let T_{lp} represent the time for the CG iterative process (i.e., *CG LP Time*). We then define $T_{CG'} = T_{lp} + T_{int}$ as the total time required for our algorithm to find its best feasible solution. In a subsequent experiment, we run CPLEX using $T_{CG'}$ (derived from the corresponding CG run) as the time limit. We then compare the best integer solution found by the time-limited CPLEX against the best integer solution obtained by our CG algorithm. The results of this comparison are presented in Table 6.

Table 6 presents the detailed results of this experiment. For the standard CPLEX, we list its final objective (*Obj*) and total *time (s)*. For the CG method, we report its *Obj*, its original *IP.gap* (relative to the standard CPLEX), its total *time (s)*, and the time required to find its best integer solution ($T_{CG'}$). The final two columns show the *Obj* achieved by CPLEX with a time limit of $T_{CG'}$ and the resulting *IP.gap with CG*. This *IP.gap with CG* is calculated by comparing the time-limited CPLEX objective to the CG objective, using the formula: *(CPLEX using $T_{CG'}$.Obj - CG.Obj) / CG.Obj × 100%*.

First, we observe in Table 6 that the time required for the CG algorithm to discover its best integer solution ($T_{CG'}$) is extremely short, averaging only 55.85 seconds. This time is a small fraction of both the full CG algorithm runtime (avg. 323.11s) and the original CPLEX runtime (avg. 1320.20s). More importantly, when CPLEX is constrained to this identical, limited runtime ($T_{CG'}$), its solution quality is demonstrably poorer. As shown in the *IP.gap with CG* column, the solutions from the time-limited CPLEX are, on average, 2.011% worse (higher objective value) than the solutions found by our CG algorithm. In six of the nine instances (e.g., Instances 22 and 23), the gap is substantial, indicating that CPLEX has not yet found a high-quality feasible solution in that short

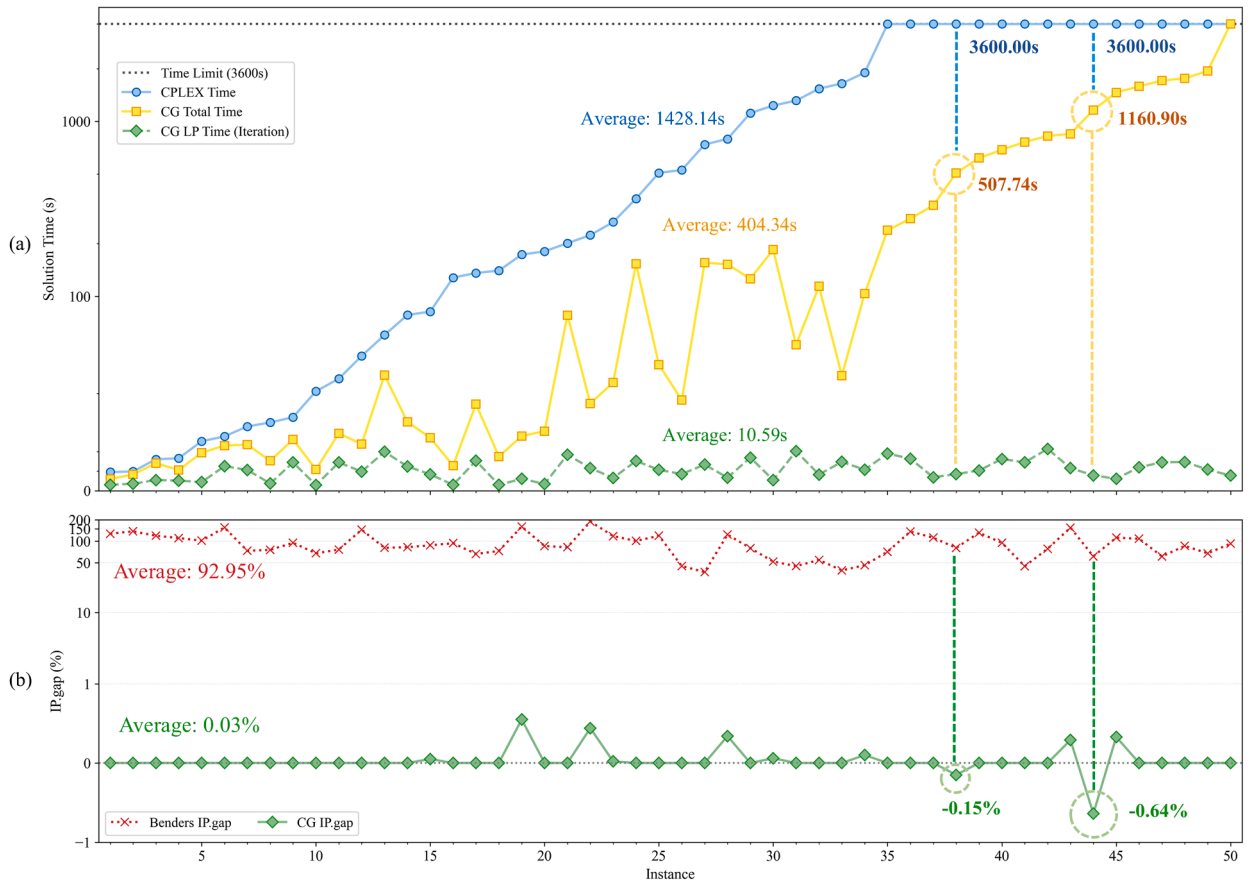

Fig. 6. Benchmark performance comparison for small-scale instances.

Table 6
Comparison of CPLEX and CG with fair time limits for specific small-scale instances.

Instance	CPLEX		CG				CPLEX using $T_{CG'}$	
	Obj	time (s)	Obj	IP.gap	time (s)	$T_{CG'}$	Obj	IP.gap with CG
15	4532209.28	92.12	4534659.32	0.05%	27.20	13.62	4660678.22	2.779%
19	3418750.07	173.25	3437675.92	0.55%	28.11	7.04	3504867.60	1.955%
22	3185639.59	223.76	3199540.01	0.44%	44.85	23.84	3398987.12	6.234%
23	4101111.97	265.66	4101843.82	0.02%	55.67	11.57	4317327.04	5.253%
28	3618362.37	793.35	3630794.49	0.34%	152.50	99.35	3642867.78	0.333%
30	5403163.99	1231.27	5406597.55	0.06%	185.11	61.15	5406338.26	-0.005%
34	5406091.74	1898.81	5411234.58	0.10%	103.77	47.92	5406091.74	-0.095%
43	4302556.59	TL	4316547.95	0.33%	1462.26	76.57	4387564.63	1.645%
45	2819569.95	TL	2827705.32	0.29%	848.49	161.62	2827705.32	0.000%
Average	4087495.06	1320.20	4096288.77	0.24%	323.11	55.85	4172491.97	2.011%

Note: TL means time limit.

timeframe. Although CPLEX find marginally better solutions in two instances (30 and 34), the average results support our conclusion. This experiment demonstrates that our proposed CG algorithm is particularly effective at rapidly identifying high-quality incumbent solutions.

In addition, as previously observed (i.e., in Fig. 6), the standard CPLEX solver fails to find optimal solutions for 16 of the more challenging instances (Instances 35–50) within the 3600-second time limit. To rigorously evaluate the actual solution quality of our proposed CG algorithm for these instances, we conduct a follow-up experiment. In this experiment, the solution obtained from our CG algorithm (within the initial 1-hour limit) is provided to CPLEX as a warm start. CPLEX is then executed with a significantly extended time limit of 24 hours (86400 seconds) to allow it to converge to the true optimal solution, or as close as possible.

The results of this comparison are presented in Table 7. We report the objective (*Obj*) and *time (s)* for the initial CG solution, alongside the final *Obj*, total *time (s)*, and resulting *IP.gap* from the 24-hour CPLEX run. The *IP.gap with CG* here is defined as *(CPLEX.Obj*

Table 7
Solution quality comparison using CG solutions as a warm start for CPLEX in small-scale instances.

Instance	CG		CPLEX with Warm Start (24h Limit)		
	Obj	time (s)	Obj	time (s)	IP.gap with CG
35	5616535.07	238.54	5616535.07	1583.87	0.00%
36	3887395.95	278.01	3887395.95	1336.25	0.00%
37	3322639.22	331.05	3322639.22	2373.21	0.00%
38	4086834.34	507.74	4086834.34	49944.08	0.00%
39	3112151.71	616.80	3112151.71	8419.50	0.00%
40	4950406.35	688.48	4950406.35	17752.71	0.00%
41	6074768.56	762.69	6074768.56	3704.93	0.00%
42	4960279.33	824.94	4960279.33	12798.06	0.00%
43	2827705.32	848.49	2819569.95	47336.62	-0.29%
44	4132371.33	1160.90	4132371.33	59862.37	0.00%
45	4316547.95	1462.26	4302556.59	3138.89	-0.32%
46	3787689.82	1586.12	3787689.82	4733.10	0.00%
47	3988652.44	1711.47	3988652.44	7830.55	0.00%
48	3970860.18	1762.81	3970860.18	22057.96	0.00%
49	3916057.58	1940.19	3916057.58	22897.38	0.00%
Average (35–49)	—	981.37	—	17717.96	-0.04%
50	3834294.93	TL	3834294.93	86401.80	0.00%

- *CG.Obj) / CG.Obj* × *100%*. Instance 50 is reported separately and excluded from the average calculation, as both the initial CG run and the subsequent CPLEX warm start run reached their respective time limits (3600s and 86400s).

The results for instances 35–49 show that the solutions found rapidly by our CG algorithm (avg. 981.37s) are already extremely close to the optimal solutions found within 24 hours. The average *IP.gap with CG* is a negligible -0.04%, confirming that our CG solutions are, on average, only 0.04% suboptimal. In fact, for 13 of the 15 instances, the solution was proven to be optimal (0.00% gap).

6.2.2. Experiments on medium- and large-scale instances

We now evaluate the performance of our proposed CG algorithm on the medium- and large-scale instances. Based on its poor performance on the small-scale instances, the Benders-based method is excluded from this comparison, and standard CPLEX is used as the sole benchmark.

For these larger problems, both methods are executed under a total time limit of 3600 seconds. However, given the immense number of potential aircraft routes, a secondary termination condition is introduced for the CG iterative process on the large-scale instances only: the column generation phase is terminated if it reached 1800 seconds (i.e., half of the total time limit). The set of aircraft routes generated up to that point is then used to construct and solve the final MIP.

Since both our CG algorithm and CPLEX consistently terminated at the 3600-second time limit, we do not compare computational time. Instead, we focus on the final solution quality. The results are presented in Fig. 7, which visualizes the complete distribution of solution gaps. The figure clearly demonstrates the superiority of our proposed algorithm. For the medium-scale instances (shown in red), the mean *IP.gap* is -4.07%. The entire interquartile range and the vast majority of individual instances lie below the 0% line, confirming that our method consistently finds better-quality solutions than CPLEX within the time limit.

This performance advantage becomes even more pronounced on the large-scale instances (shown in blue). The mean *IP.gap* improves significantly to -6.32%. This result shows that as the problem scale and complexity increase, the performance gap between our CG algorithm over CPLEX widens. While CPLEX struggles with the larger problem space, our algorithm is able to identify high-quality solutions more effectively. The figure also highlights the full range of outcomes, including extreme instances where our method found solutions more than 14% (Medium) and 15% (Large) better than those found by CPLEX.

6.2.3. Impact of the TUM property

To quantify the computational advantage gained from exploiting the model's TUM property, we conduct an experiment on the 34 small-scale instances that CPLEX solved to optimality within the 3600-second limit. The primary model used throughout this paper (denoted as *CPLEX*) leverages this property by relaxing the integer constraints on the flow-related variables (e.g., $\boldsymbol{y}$, $\boldsymbol{\phi}$, and $\boldsymbol{z}$). As a comparative benchmark, we define a *Standard MIP* where this property is not exploited, and these same variables are explicitly defined as integers.

Table 8 presents the results of this comparison. The *CPLEX.time (s)* on average (avg. 405.12s) is significantly faster than the *Standard MIP. time(s)* (avg. 576.34s). The final column, *Time Difference (%)*, quantifies the percentage change in computation time relative to our model, calculated as *(Standard MIP.time(s) - CPLEX.time(s)) / CPLEX.time(s)* × *100%*.

The results show that the *Standard MIP* required an average of 78.73% more computational time to solve the same instances. Although the *Standard MIP* was unexpectedly faster in two instances (26 and 33), the overwhelming trend confirms the benefit. For

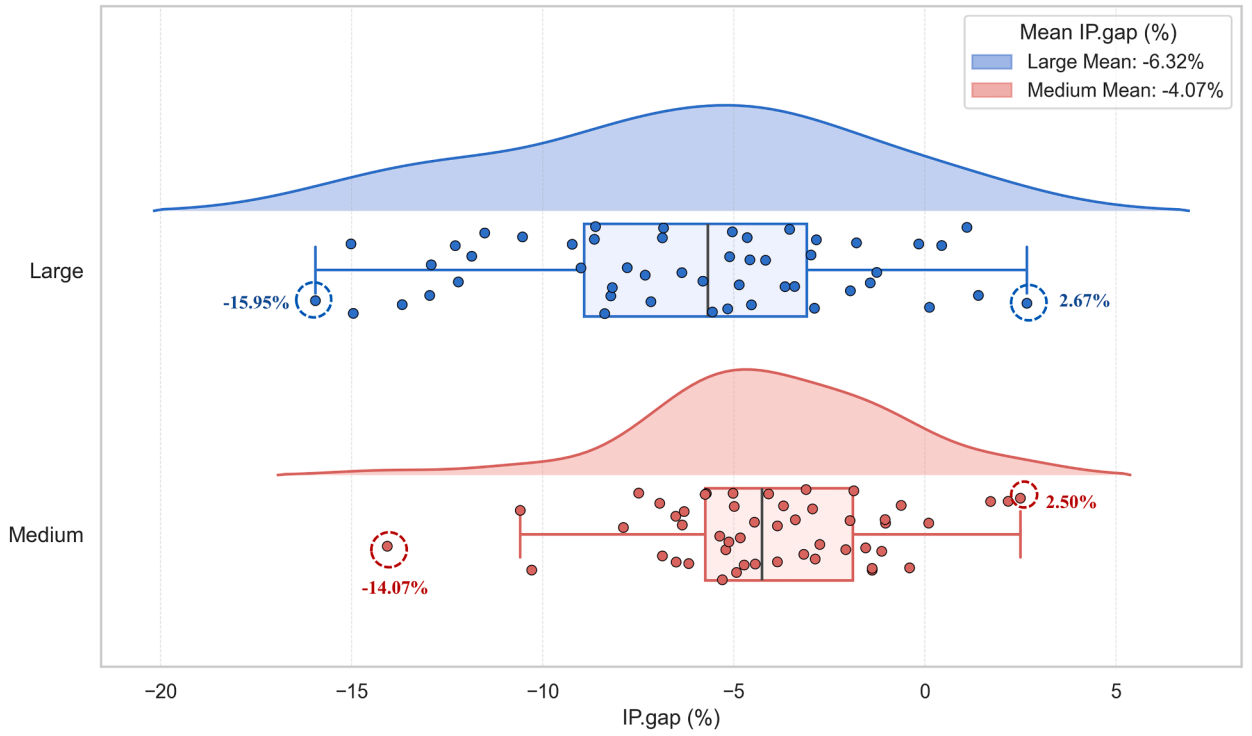

Fig. 7. IP Gap distribution for medium- and large-scale instances.

many instances (e.g., 2, 4, 5, 22, 23), the time saving is substantial. This analysis validates our modeling choice, demonstrating that exploiting the TUM property provides a significant computational advantage by simplifying the problem for the MIP solver.

6.2.4. Impact of tightening constraints

In Section 5.1.1, we introduce a set of tightening constraints (4), for the cargo flight route flow variables in the RMP. To evaluate the effectiveness of these constraints, we compare the performance of our full CG algorithm (which includes them) against a variant where these constraints are removed from the RMP.

The results presented in Table 9 demonstrate the critical importance of these constraints. On average, the CG algorithm with tightening constraints solves the instances in 404.34 seconds, which is more than twice as fast as the 868.13 seconds required by the variant without them. This confirms that constraints (4) are highly effective in strengthening the RMP formulation.

6.3. Value of through cargo connections

As discussed in the introduction, a primary contribution of this study is to evaluate the value of through cargo connections in the air cargo operations. This section now presents the detailed numerical results to quantify this impact. As mentioned previously, the 50 small-scale instances form a reliable basis for our analysis. For these instances, 34 instances of them are solved to optimality by CPLEX within one hour, and another 15 are solved to optimality within 24 hours. For the remaining instance, we obtain a solution with an optimality gap of 0.07%, indicating a solution very close to the optimal. We therefore use these 50 (near-)optimal solutions to analyze the impact of through cargo connections.

We compare the solutions obtained for different versions of model (3). We let **Model O** denote a variant of model (3) that does not allow performing any through cargo connection. We let **Model TH** denote a variant of model (3) that does not allow performing through cargo connections at non-hub airports. Specifically, in **Model TH**, we assume that each intermediate stop of a through cargo connection needs to be executed at a hub airport. We let **Model TN** refer to the complete problem, with through cargo connections allowed in any airport. In Appendix E, we provide a detailed mathematical representation of the implementation of **Model O** and **Model TH**. To perform this comparison, the other two models (**Model O** and **Model TH**) are also solved to optimality using CPLEX. The solutions produced by the three models are compared in Fig. 8.

Fig. 8 clearly demonstrates the significant benefits of allowing flexible through cargo connections. As for the overall operational cost (i.e., Objective Value), the flexibility to perform connections has a striking impact on the total objective value. As shown in Fig. 8(a), **Model O** (no through cargo connections) consistently results in the highest-cost solutions. **Model TH** (hub-only through cargo connections) offers a slight improvement. However, **Model TN** (through cargo connections at any airport) achieves a dramatically lower objective value in nearly every instance. On average, **Model TN** improves the objective by approximately 14.08% compared to **Model O** and 11.26% compared to **Model TH**.

Table 8
Comparison of solution times with and without exploiting the TUM property in small-scale instances.

Instance	CPLEX.time(s)	Standard MIP.time(s)	Time Difference (%)
1	9.58	23.68	147.20%
2	9.86	31.00	214.31%
3	16.16	32.75	102.67%
4	16.67	50.15	200.88%
5	25.42	77.20	203.65%
6	27.92	48.90	75.16%
7	33.05	34.08	3.12%
8	35.07	100.64	186.96%
9	37.82	52.68	39.28%
10	51.13	58.74	14.88%
11	57.61	83.05	44.16%
12	69.29	164.31	137.14%
13	80.07	113.76	42.07%
14	90.40	140.77	55.72%
15	92.12	104.05	12.95%
16	127.84	158.14	23.71%
17	135.96	165.91	22.03%
18	140.22	152.74	8.93%
19	173.25	264.47	52.65%
20	180.69	217.18	20.20%
21	201.24	362.95	80.35%
22	223.76	744.21	232.60%
23	265.66	855.15	221.90%
24	361.79	580.76	60.53%
25	507.81	829.36	5.99%
26	527.11	1289.78	144.69%
27	737.57	643.25	-12.79%
28	793.35	1923.53	142.46%
29	1115.70	1706.97	53.00%
30	1231.27	2196.60	78.40%
31	1314.24	1618.90	23.18%
32	1537.94	1743.66	13.38%
33	1647.74	1087.13	-34.02%
34	1898.81	1939.18	2.13%
Average	405.12	576.34	78.73%

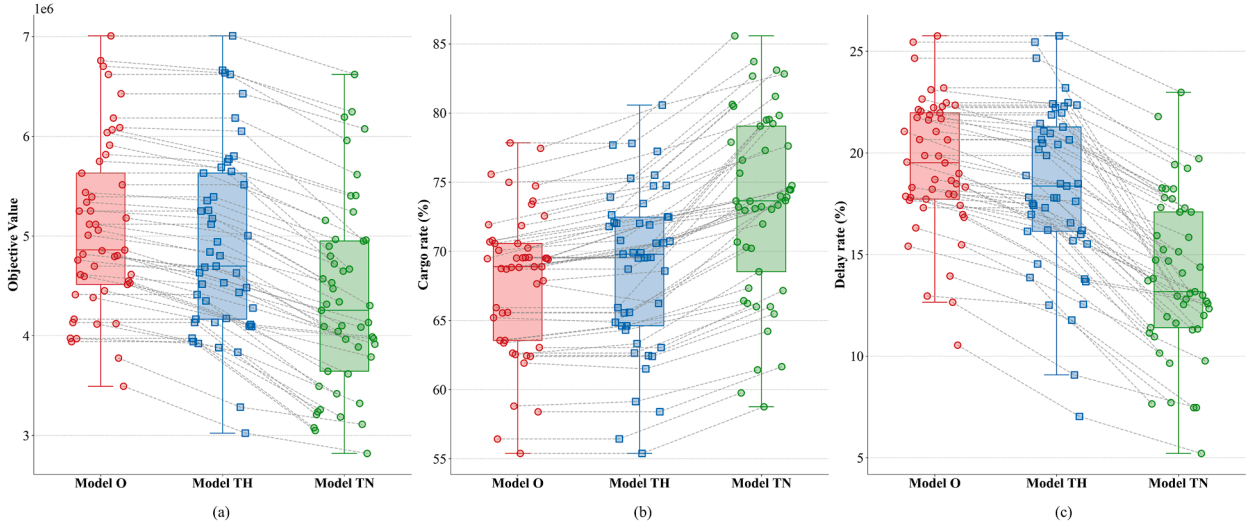

Fig. 8. Impact of through cargo connection on the operational cost and service levels.

Table 9
Impact of tightening constraints on CG performance in small-scale instances.

Instance	CG with Tighten Constraints				CG without Tighten Constraints			
	Obj	time (s)	CG.time (s)	CG.Num	Obj	time (s)	CG.time (s)	CG.Num
1	3077508.64	5.99	3.04	51	3077508.64	13.11	5.52	53
2	3049679.42	8.37	3.56	54	3097724.14	19.74	5.12	48
3	3209616.42	14.11	5.48	58	3209616.42	15.24	5.69	54
4	3237875.09	10.64	5.18	55	3237875.09	13.19	5.12	50
5	3494104.51	19.50	4.39	74	3494104.51	13.82	3.59	63
6	3258069.05	23.21	12.60	46	3258069.05	24.21	12.59	49
7	4255750.67	23.67	10.68	64	4255750.67	24.82	10.27	59
8	4571077.40	15.39	3.71	63	4571077.40	59.96	6.07	53
9	4092144.21	26.38	14.52	57	4092144.21	21.11	10.22	42
10	5157459.50	10.86	2.88	53	5157459.50	39.01	6.59	63
11	4316882.68	29.47	14.51	53	4316882.68	34.56	13.11	55
12	3644077.18	23.97	9.81	49	3656425.18	37.15	12.62	57
13	4899055.76	59.48	20.00	72	4899055.76	55.77	22.87	69
14	4798325.10	35.40	12.49	76	4798325.10	31.12	12.93	74
15	4534659.32	27.20	8.39	52	4534659.32	38.17	13.63	58
16	4472921.05	12.91	3.01	51	4472921.05	48.50	5.06	50
17	4721504.55	44.59	15.35	57	4784866.15	50.26	18.27	61
18	4966981.28	17.57	3.02	51	4966981.28	88.38	6.79	64
19	3437675.92	28.11	6.08	52	3429128.87	23.93	6.29	55
20	4043901.03	30.63	3.48	57	4043901.03	80.07	6.99	58
21	4340854.27	90.30	18.50	65	4340854.27	112.82	23.10	70
22	3199540.01	44.85	11.65	56	3199540.01	64.99	11.49	54
23	4101843.82	55.67	6.54	60	4101111.97	174.82	9.85	51
24	4648230.43	153.67	15.29	60	4648230.43	396.10	17.70	64
25	6193499.27	64.83	10.72	68	6202662.61	282.94	20.83	79
26	3965332.87	46.58	8.57	62	3965332.87	200.83	12.22	59
27	5959763.00	155.59	13.40	68	6048247.36	493.92	18.00	65
28	3630794.49	152.50	6.72	56	3618362.37	192.23	13.62	65
29	4669026.45	126.05	17.02	65	4669026.45	210.01	17.72	63
30	5406597.55	185.11	5.48	56	5406597.55	248.73	10.55	53
31	6246092.08	75.10	20.30	73	6246092.08	345.03	22.47	80
32	5242828.93	114.07	8.30	51	5242828.93	122.07	5.28	50
33	6621099.63	59.21	14.79	58	6621099.63	137.64	12.15	72
34	5411234.58	103.77	10.78	53	5406091.74	239.24	10.02	49
35	5616535.07	238.54	19.12	71	5616535.07	1890.02	24.80	78
36	3887395.95	278.01	16.42	59	3887395.95	736.98	20.41	61
37	3322639.22	331.05	6.87	46	3322639.22	337.28	13.40	49
38	4086834.34	507.74	8.55	54	4086834.34	3091.35	14.70	56
39	3112151.71	616.80	10.39	49	3189275.33	3602.56	17.92	63
40	4950406.35	688.48	16.30	60	4950406.35	3603.12	16.14	59
41	6074768.56	762.69	14.58	69	6074768.56	1227.97	19.56	63
42	4960279.33	824.94	21.49	81	4960279.33	2075.83	19.21	70
43	2827705.32	848.49	11.63	49	2819569.95	2156.39	10.10	39
44	4132371.33	1160.90	7.86	54	4153153.15	3603.16	13.35	49
45	4316547.95	1462.26	6.15	56	4313652.66	1107.00	11.40	53
46	3787689.82	1586.12	12.04	83	3787689.82	2243.10	21.25	73
47	3988652.44	1711.47	14.62	96	4028921.50	2969.97	19.24	73
48	3970860.18	1762.81	14.75	103	3970860.18	3602.10	23.78	95
49	3916057.58	1940.19	10.85	46	3916057.58	3603.09	14.15	52
50	3834294.93	3601.75	7.76	50	3844922.51	3603.32	13.05	46
Average	—	404.34	10.59	60.44	—	868.13	13.34	59.76

Regarding service quality, we define *Cargo rate (%)* represents the volume transported by cargo aircraft divided by the total cargo demand. We define the *Delay rate (%)* as the ratio of the total cargo volume that cannot be transported within its time window to the total cargo demand. The improved objective value of **Model TN** is not a simple cost trade-off, but also a more efficient network operation that enhances service quality. Specifically, for the *Cargo rate (%)*, Fig. 8(b) shows that the added flexibility allows the network to carry significantly more cargo. The average cargo rate for **Model TN** (73.47%) is substantially higher than that of **Model O** (67.98%) and **Model TH** (68.90%). Most importantly, in terms of the *Delay rate (%)*, the operational flexibility of **Model TN** drastically reduces cargo delays. Fig. 8(c) shows that the average delay rate is reduced to 13.85% in **Model TN**, a significant drop from 19.38% in **Model O** and 18.23% in **Model TH**.

In summary, the comparison reveals that while restricting through cargo connections to hub airports (**Model TH**) provides a marginal benefit, the value of through cargo connections lies in the flexibility to perform such connections at any airport in the

Table 10
Detailed analysis on trips with different traveling times.

Instance	Penalty	T_1				T_2				T_3			
		Belly rate	Delay rate	Cargo rate	Belly.util	Belly rate	Delay rate	Cargo rate	Belly.util	Belly rate	Delay rate	Cargo rate	Belly.util
S10	5	19.07%	21.72%	59.21%	59.81%	12.46%	20.75%	66.79%	58.69%	12.81%	57.45%	29.74%	88.36%
	10	21.84%	21.72%	56.44%	68.49%	15.35%	15.46%	69.19%	72.30%	14.49%	52.37%	33.13%	100.00%
	20–70	17.38%	17.03%	65.58%	54.53%	14.39%	12.69%	72.92%	67.77%	14.49%	57.45%	28.05%	100.00%
S14	5	5.13%	14.36%	80.51%	71.71%	12.22%	26.26%	61.52%	68.22%	11.04%	69.89%	19.08%	86.53%
	10–50	5.13%	15.39%	79.48%	71.71%	10.99%	16.29%	72.73%	61.31%	10.18%	28.12%	61.71%	79.80%
	60–70	5.13%	8.58%	86.29%	71.71%	12.67%	18.94%	68.39%	70.72%	11.04%	29.92%	59.04%	86.53%
S25	5	12.56%	14.99%	72.44%	58.72%	15.37%	26.05%	58.58%	74.40%	24.01%	31.12%	44.87%	67.24%
	10	12.56%	11.66%	75.77%	58.72%	16.60%	20.71%	62.69%	80.37%	34.57%	25.86%	39.57%	96.78%
	20	16.96%	9.06%	73.98%	79.25%	16.22%	20.58%	63.20%	78.51%	34.57%	24.19%	41.24%	96.78%
	30–50	11.72%	6.98%	81.31%	54.77%	16.56%	19.64%	63.81%	80.13%	34.57%	27.05%	38.39%	96.78%
	60–70	11.72%	6.98%	81.31%	54.77%	15.45%	19.44%	65.11%	74.78%	34.57%	27.05%	38.39%	96.78%
S34	5	5.27%	8.29%	86.43%	40.45%	12.30%	30.25%	57.46%	67.43%	34.36%	33.59%	32.05%	86.73%
	10	5.77%	14.88%	79.35%	44.27%	12.29%	16.23%	71.48%	67.39%	39.62%	40.03%	20.34%	100.00%
	20	5.61%	6.04%	88.35%	43.00%	12.81%	16.28%	70.91%	70.21%	39.62%	55.06%	5.32%	100.00%
	30	5.26%	16.36%	78.38%	40.33%	13.11%	16.31%	70.58%	71.89%	39.62%	33.60%	26.78%	100.00%
	40–70	6.29%	16.36%	77.35%	48.25%	13.71%	16.06%	70.23%	75.17%	39.62%	34.85%	25.52%	100.00%

Note: A penalty range (e.g., "20-70") indicates that the results are identical for all discrete penalty values in that range (i.e., 20, 30, …, 70). These rows have been consolidated for conciseness.

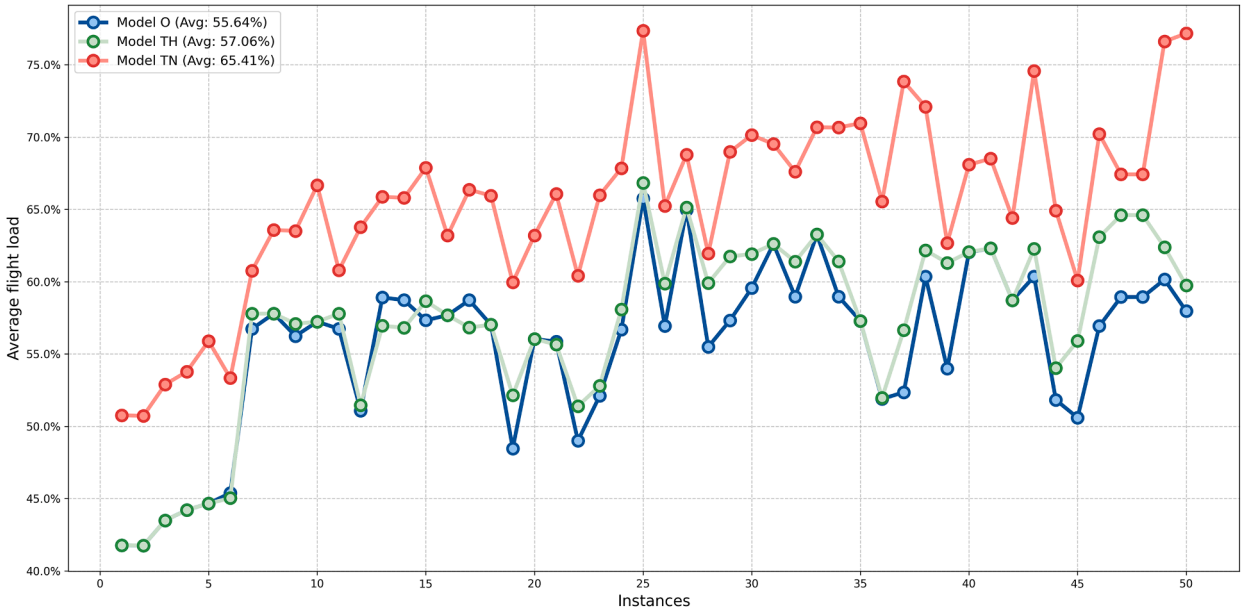

Fig. 9. Impact of through cargo connection on average flight load.

network (**Model TN**). This flexibility allows the model to find superior cargo routing and consolidation strategies, simultaneously lowering overall costs, increasing the amount of transported cargo, and significantly reducing delivery delays.

Besides, to evaluate the impact of through cargo connection on operational efficiency, we compare the average flight load factor achieved by the three model variants. The average flight load represents the ratio of the total fulfilled load to the total available capacity across all assigned flights. Fig. 9 illustrates the performance of **Model O**, **Model TH**, and **Model TN** across all 50 small-scale instances.

First, we can observe a marginal but consistent improvement when moving from **Model O** to **Model TH**. As depicted by the closely coupled blue and green plots, **Model TH** consistently outperforms **Model O** on the majority of instances. This small performance gain (approx. 1.42% on average) reveals that the introduction of through cargo connections provides a tangible limited benefit by shortening cargo transshipment time, even when strictly restricted to designated hub airports. Second, by relaxing the constraint that through cargo connections must occur at hubs, **Model TN** is permitted to perform through cargo connections at any airport in the network. This added operational flexibility can yield a significant 8.35% increase in the average load factor compared to the constrained **Model TH**.

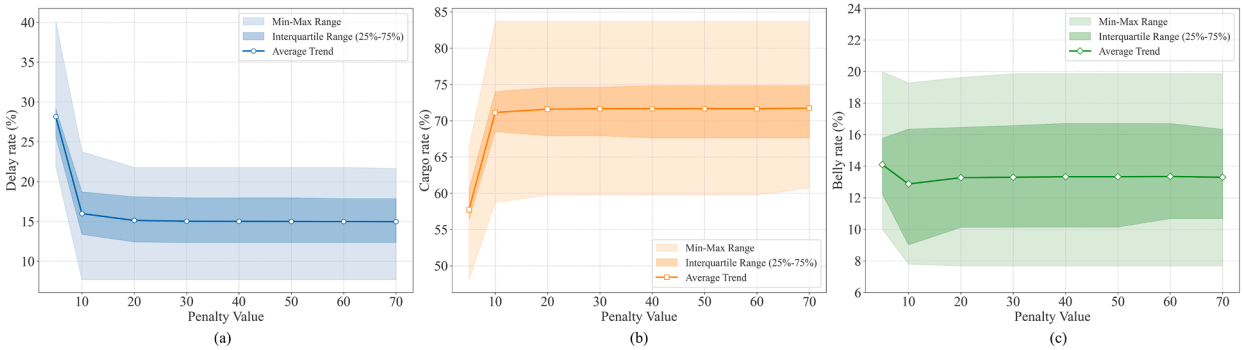

Fig. 10. Three key performance indicators under different penalty values.

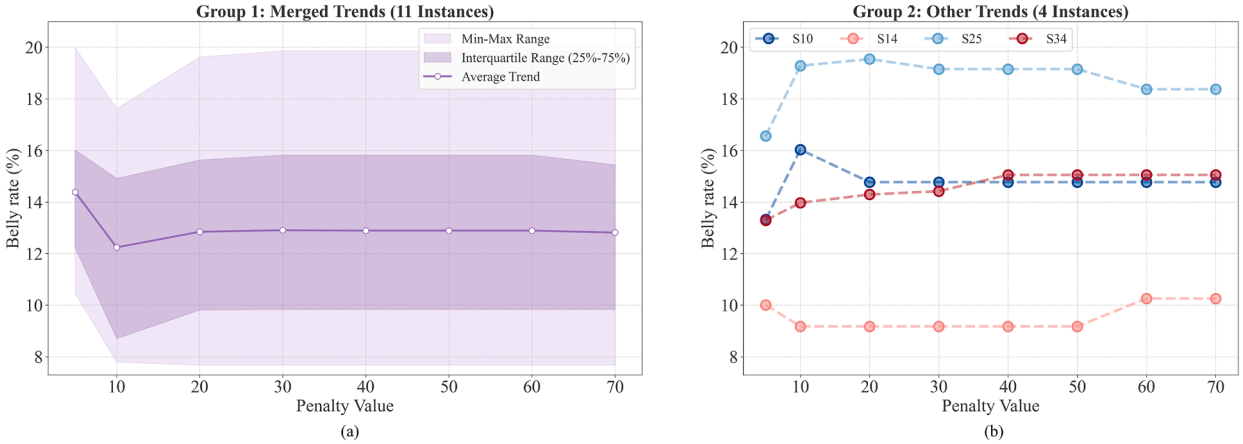

Fig. 11. Belly rate under different penalty values.

In summary, while through cargo connections (**Model TH**) offers a slight advantage over a no-connection policy (**Model O**), it is the flexibility of these connections (**Model TN**) that is the dominant factor in maximizing the utilization of the cargo fleet. This observation emphasizes the advantage of through cargo connections in facilitating efficient cargo transportation, both in terms of time and space.

6.4. Impact of penalties for unserved demands

In this section, we analyze the impact that unserved demand penalties have on the solutions produced. We set the value of the unserved demand penalty η^q to values from the set $\{5, 10, 20, 30, 40, 50, 60, 70\}$. Our experiments are conducted based on the small-scale instances using CPLEX. To ensure a fair comparison of the solution structure, we select a subset of 15 instances for which CPLEX can obtain a proven optimal solution within one hour for all considered penalty values.

The aggregated trends of three key performance indicators for these 15 instances are presented in Fig. 10. In this context, the *Delay rate (%)*, also referred to as the unserved demand ratio, represents the total volume of cargo that cannot be transported within its time window divided by the total cargo demand. The *Cargo rate (%)* represents the volume transported by cargo aircraft divided by the total cargo demand. Finally, the *Belly rate (%)* denotes the volume transported via passenger belly capacity divided by the total cargo demand.

From the figure, we can observe distinct responses for each indicator. When the penalty is low (i.e., $\eta^q = 5$), the model prefers to incur high delay rates rather than high transportation costs. Once the penalty increases to 10, the model's strategy fundamentally shifts, causing the average *Delay rate (%)* to plummet and the average *Cargo rate (%)* to rise sharply. Subsequently, for all $\eta^q \geq 10$, these two metrics become exceptionally stable. Increasing the penalty further has a negligible marginal effect, and the interquartile range (IQR) for both metrics compresses, indicating robust and consistent solutions across all instances.

In contrast, the *Belly rate (%)* exhibits a more complex and relatively unstable trend. The average trend decreases from $\eta^q = 5$ to $\eta^q = 10$ before slightly recovering and stabilizing, and its IQR remains wide across all penalty values. This suggests that the use of belly capacity is highly instance-specific and that aggregating all instances into a single trend may obscure distinct underlying behaviors. We therefore focus on analyzing the individual trends of the *Belly rate (%)*, which can be broadly categorized into two main groups, as illustrated in Fig. 11.

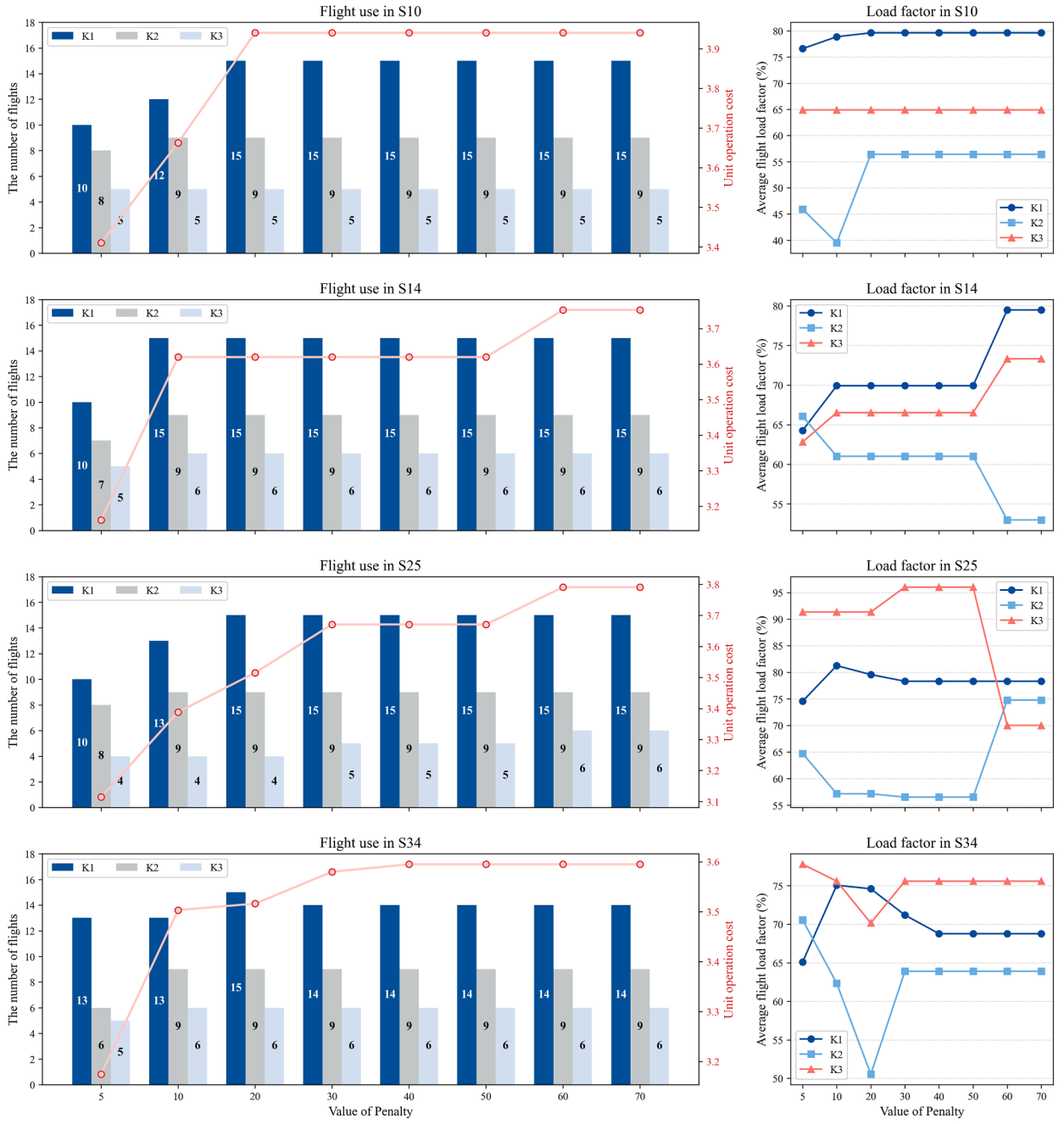

Fig. 12. Flight use details in specific instances.

As illustrated in Fig. 11, the belly rate trends for instances S10, S14, S25, and S34 (Group 2) diverge significantly from the aggregated trend of the other 11 instances (Group 1). Therefore, we make a more detailed analysis on those four instances. We first segment the cargo demand based on direct flight time into three categories: short-haul (T_1), corresponding to the (0, 120] minute window, medium-haul (T_2), for the (120, 180] minute window, and long-haul (T_3), for the (180, 540] minute window. We then analyze the proportion of demand in each time segment fulfilled by different transportation modes: belly capacity (*Belly rate*), delayed shipment (*Delay rate*), and dedicated cargo aircraft (*Cargo rate*). Furthermore, we define belly utilization (*Belly.util*) as the ratio of used belly capacity to the total available belly capacity within each time segment. The detailed results for these metrics, as a function of the penalty value, are presented in Table 10. Meanwhile, a detailed analysis of the flight operations for these four instances is provided in Fig. 12.

As observed in Table 10, the model strategically allocates cargo demand by haul length: dedicated cargo aircraft predominantly serve short (T_1) and medium-haul (T_2) cargo demand, evidenced by high cargo rates in these segments. Conversely, long-haul (T_3) cargo demand is primarily fulfilled by belly capacity or delayed. This is a logical strategy, as using a cargo aircraft for a long-haul T_3 route would consume its entire operational window, incurring a high opportunity cost. Therefore, the model reserves this expensive T_3 demand for belly capacity, which acts as a valuable supplement. This is confirmed by the T_3 *Belly.util* (i.e., Table 10), which approaches 100% in most cases as the penalty value increases. The S34 instance, transitioning from penalty 10 to 20, perfectly illustrates that the T_3 cargo rate drops significantly as this cargo demand is offloaded, but the total number of flights (Fig. 12) increases to improve service in the T_1 segment.

The fleet operation data in Fig. 12 provides further insight into the cost-versus-service trade-off. A critical observation, consistent across all four instances, is the rise in the "Unit operating cost" as the delay penalty increases, even when the flight count remains stable. This metric is defined as the sum of total fixed and variable operational costs for the cargo fleet, divided by the total cargo volume transported by those aircraft. This result demonstrates that as penalties rise, the model prioritizes service (delay reduction) over fleet efficiency.

This trade-off is most evident in the load factor subfigure. The K1 fleet acts as the stable workhorse of the network, while the K3 fleet consistently maintains exceptionally high load factors (e.g., more than 90% in S25), supporting that its high fixed cost necessitates high utilization to be economical. The most nuanced behavior is seen in the K2 fleet, which functions as a flexibility buffer. In instances S14 and S34, the K2 load factor becomes highly volatile, dropping as low as 50-55%. This indicates that the model is willing to deploy type-K2 flights at very low efficiency to prevent incurring an even more expensive delay penalty.

7. Conclusions and perspectives

This study focuses on air cargo express operations. It addresses an air cargo service network scheduling problem aimed at determining flight schedules, dedicated cargo aircraft routes, and air cargo allocation. The study integrates the transportation capabilities of dedicated cargo aircraft and rented belly capacity from passenger airlines while incorporating through cargo connections within cargo transshipments. We propose an integrated route-based mathematical formulation and investigate its TUM property. We introduce a column-generation-based heuristic to manage the exponential number of potential aircraft routes. To initialize the process, we construct the initial restricted master problem by incorporating tightened constraints on cargo flight route variables. In the pricing problem, we formulate the aircraft route generation for each aircraft type as a longest path identification problem on a directed acyclic graph. We then propose a Δ-longest path finding algorithm to efficiently identify negative reduced cost variables. Finally, we construct a MIP model that includes all aircraft routes generated during the CG process, as well as regenerated aircraft routes derived from partial flight slots and duplicated aircraft types. The cargo routes are then regenerated according to this final set of aircraft routes. This MIP, which represents a restriction of the original problem, is solved to produce high-quality solutions in shorter times.

We conduct computational experiments to validate the performance of our proposed algorithm. On 50 small-scale instances, our CG algorithm is compared with CPLEX and a Benders-based method. The CG algorithm outperforms the Benders-based method in both solution quality and computational time. Compared to CPLEX, our CG algorithm demonstrates strong performance, as its *IP.gap* remains at 0.00% for most instances, averages a negligible value of 0.03%, and requires lower average computational times. Further analysis demonstrates that our CG algorithm is particularly effective at rapidly identifying high-quality incumbent solutions. For 15 instances not solved to optimality by CPLEX within the initial time limit, extending the limit to 24 hours and providing the CG solution as a warm start allows CPLEX to prove optimality for 13 instances, resulting in a negligible average optimal gap of 0.04%. On medium- and large-scale instances, our CG algorithm maintains strong performance. It finds solutions with an average *IP.gap* of -4.07% (medium) and -6.32% (large) relative to the solutions found by CPLEX within the time limit. In some instances, the solution quality improvement exceeds 14% (medium) and 15% (large). Finally, component analysis on small-scale instances demonstrates that exploiting the TUM property yields a computational advantage. The inclusion of flow tightening constraints also reduces the average solution time by more than 50%.

In our managerial analysis, we investigate the impact of through cargo connections, observing that their utilization significantly reduces delayed demand and increases flight load factors by avoiding transhhipment operations. We also examine the imapct of the delayed demand penalty. As the penalty value increases, delayed demand becomes concentrated on long-haul segement, while aircraft capacity is prioritized for short-haul and medium-haul cargo demand. Belly capacity effectively supplements dedicated aircraft capacity on these long-haul segement. This trade-off is also reflected in fleet-specific load factors. The K1 fleet acts as the stable base capacity for the network. The K3 fleet, which has high fixed costs, consistently maintains high load factors to ensure economic viability. The K2 fleet functions as a flexibility buffer. In some instances, its load factor drops significantly, indicating the model deploys type-K2 flights at low utilization to prevent incurring even higher penalty costs. These results highlight the dynamics carriers must consider when making decisions about fleet utilization and operational cost structures.

The results of those experiments indicate that the belly capacity of passenger flights can effectively supplement dedicated cargo aircraft. However, the belly capacity for a passenger flight is typically revealed close to its departure, following the full disclosure of passenger and baggage information. This results in uncertainty regarding the availability of capacity, at the decision-making stage. A prospective direction for future research is to incorporate the uncertainty associated with belly capacity and cargo handling into air cargo service network scheduling problems, thereby enhancing the robustness of the network. Besides, the timeliness of cargo delivery is a key factor that can be further integrated into the optimization of air cargo service networks. Considering varying arrival times correspond to different revenue levels to highlight the time-sensitive nature of cargo deliveries, this perspective can significantly enhance the efficiency and profitability of the air cargo service network.

CRediT authorship contribution statement

Ling Zhu: Writing – original draft, Visualization, Conceptualization; **Simon Belieres:** Writing – review & editing, Validation; **Mike Hewitt:** Writing – review & editing, Investigation; **Lingxiao Wu:** Writing – review & editing, Supervision, Methodology.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We are grateful to the editor and two anonymous reviewers for their valuable comments, which helped improve the quality of the paper. This work was supported by the National Natural Science Foundation of China under Grant 72301230, the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant 25223223, and by the Shenzhen Science and Technology Program, China, under Grant JCYJ20240813162012016.

Appendix A. Cargo route generation algorithm

Algorithm 3 Cargo Route Generation.

Input: Q: Set of cargo demands; L: Set of flight slots; K: Set of aircraft types;
H: Set of hub airports; b: Maximum path length; t_{sc}: Standard cargo transfer time.
Output: R: Set of valid cargo routes.
procedure GenRoutes(Q, L, K, b, H, t_{sc})
  Get flight operation set: $LK \leftarrow L \times K$.
  Initialize candidate cargo routes set: $CR \leftarrow \emptyset$.
  Initialize valid cargo routes set: $R \leftarrow \emptyset$.
  for each cargo $q \in Q$ **do**
    $CR_q \leftarrow$ FindPaths(q, LK, b);
    $CR \leftarrow CR \cup CR_q$
  for each route $r = [(l_1, k_1), \ldots, (l_n, k_n)] \in CR$ **do**
    isValid $\leftarrow$ true
    for $i = 1$ **to** $n - 1$ **do**
      conn $\leftarrow ((l_i, k_i), (l_{i+1}, k_{i+1}))$
      if not IsThrough(conn) **and not** IsStandard(conn, H, t_{sc}) **then**
        isValid $\leftarrow$ false; **break**
    if isValid **then**
      $R \leftarrow R \cup \{r\}$
  return R.
procedure FindPaths(q, LK, b)
  Initialize set: $CR_q \leftarrow \emptyset$.
  for each operation $(l, k) \in LK$ **do**
    if $o^l = o^q$ **and** $st^l \geq st^q$ **then**
      path $\pi \leftarrow [(l, k)]$; Extend(q, LK, b, π, CR_q).
  return CR_q.
procedure Extend(q, LK, b, π, CR_q)
  $(l_{last}, k_{last}) \leftarrow$ last element of π
  if $d^{l_{last}} = d^q$ **and** $et^{l_{last}} \leq et^q$ **then**
    Update set: $CR_q \leftarrow CR_q \cup \{\pi\}$.
  if length(π) $\geq b$ **then**
    return
  for each operation $(l_{next}, k_{next}) \in LK$ **do**
    if $d^{l_{last}} = o^{l_{next}}$ **and** $et^{l_{last}} \leq st^{l_{next}}$ **then**
      new path $\pi' \leftarrow \pi + [(l_{next}, k_{next})]$; Extend($q, LK, b, \pi', CR_q$).
function IsThrough(conn)
  $((l_i, k_i), (l_{i+1}, k_{i+1})) \leftarrow$ conn
  if $k_i = k_{i+1}$ **then**
    return true.
  return false.
function IsStandard(conn, H, t_{sc})
  $((l_i, k_i), (l_{i+1}, k_{i+1})) \leftarrow$ conn
  if $d^{l_i} \in H$ **and** $st^{l_{i+1}} - et^{l_i} \geq t_{sc}$ **then**
    return true.
  return false.

Appendix B. Proof of Theorem 1

Proof. Initially, we examine the composition of the constraint (3h). Define (q_{ji}, b_j) to indicate that the cargo demand q_{ji} can be transported by the belly $b_j, \forall b_j \in B$. Consequently, the constraint coefficient matrix is represented as matrix (B.1).

$$\begin{array}{c} \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (q_{11},b_1) & \cdots & (q_{1n},b_1) & (q_{21},b_2) & \cdots & (q_{2n},b_2) & \cdots & (q_{m1},b_m) & \cdots & (q_{mn},b_m) \end{array} \\ \begin{pmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 1 & \cdots & 1 \end{pmatrix} \end{array} \tag{B.1}$$

It is clear that each column in the matrix will have exactly one $+1$ element. Therefore, for any subset of rows C in matrix (B.1), one can always divide it into two disjoint sets, C_1 and C_2, such that for every column, the difference between the sum of the rows in C_1 and the sum of the rows in C_2 produces a vector with entries only 0, +1, or −1, which adheres to the Ghouila-Houri characterization.

Constraints (3h) and (3j) exhibit a comparable structure, as for each row, given the indices q and b (constraint (3h)) or the indices (r, l_m, l_n, k) and q (constraint (3j)), the related variables with these indices will only appear in the respective row. These constraints can also be represented by a matrix similar to matrix (B.1). Therefore, these constraints naturally hold for the Ghouila-Houri characterization.

Then, we analyze the structure of the constraint (3i). Considering there is a cargo demand $q_1 \in Q$ is transported using a cargo route r_1 with two through cargo connections (r_1, l_1, l_2, k) and (r_1, l_2, l_3, k) transported by the type-k aircraft, the matrice related with $y_{r_1}^{q_1}$ in constraint (3i) can be shown as belows:

$$\begin{array}{c} \\ (q_1,(r_1,l_1,l_2,k)) \\ (q_1,(r_1,l_2,l_3,k)) \end{array} \begin{array}{c} \begin{array}{ccc} \gamma^{q_1}_{(r_1,l_1,l_2,k)} & \gamma^{q_1}_{(r_1,l_2,l_3,k)} & y^{q_1}_{r_1} \end{array} \\ \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix} \end{array} \tag{B.2}$$

For other indices q, and (r, l_m, l_n, k), variables $\gamma^{q_1}_{(r_1,l_1,l_2,k)}$, $\gamma^{q_1}_{(r_1,l_2,l_3,k)}$, and $y^{q_1}_{r_1}$ do not appear in these constraints. Consequently, in the matrix corresponding to constraint (3i), the elements in the remaining rows of the columns associated with these variables will be zero. Likewise, for the indices q_1, (r_1, l_1, l_2, k), and (r_1, l_2, l_3, k), other variables $\gamma^{q}_{(r,l_m,l_n,k)}$ and y^{q}_{r} are absent in these rows, indicating that the corresponding elements are zero. Subsequently, a straightforward method is provided below to partition the matrix associated with constraint (3i) to fulfill the Ghouila-Houri characterization. For any subset of columns S in the matrix, we can easily assign all columns from this subset to one set S_1, while another set, S_2, remains empty. As noted previously, for each row, the difference between the sum of columns in S_1 and the sum of columns in S_2 results in 0, +1, or −1.

Finally, we examine the structure of constraint (3g). There are four different types of cargo routes in those constraints as below.

- Type 1: Direct routes, which indicate that the cargo route consists of only one single flight.
- Type 2: Routes with two transshipments, making the cargo route consist of three flights.
- Type 3: Routes involving one transshipment that can be a segment of a Type 2 route. For instance, if a Type 2 route is $l_1 - l_2 - l_3$, then a Type 3 route would be $l_1 - l_2$.
- Type 4: Routes with a single transshipment but cannot be considered part of Type 2 routes.

Considering a subset of columns in the matrix related to the constraint (3g), we initially divided these columns into four matrix based on the characterization of the four types mentioned above. For Type 1 martix, in a specific constraint row, such as flight f_k^l, only one cargo route containing this flight will appear. This implies that the sum of rows in each column would be 1 and the sum of columns in each row is also 1. Thus, we can easily divide the rows in the Type 1 matrix into two distinct sets such that the difference in the sum of the rows in each column between these sets would be $+1$ or -1.

For the Type 2 matrix, the cargo routes consist entirely of three flights. We can categorize the flights in this matrix into three sets: the first-leg set, the second-leg set, and the last-leg set. Using a straightforward algorithm, we identify a flight as part of the first-leg set if it serves as the first leg on one cargo route and also appears as the first leg on other cargo routes. Similarly, we can identify flights for the last-leg set. The remaining flights in this matrix are categorized into the second-leg set. Then, we can easily divide the rows in this matrix into two disjoint sets: one set, S_{21}, contains rows corresponding to the first-leg and last-leg sets, while the other set, S_{22}, includes rows corresponding to the second-leg set. These two sets satisfy the Ghouila-Houri characterization.

For the Type 3 matrix, the process is analogous to that of the Type 2 matrix, allowing it to be divided into three sets of legs. The rows associated with the first-leg set and the last-leg set are grouped into one set, denoted as S_{31}, while the remaining rows are placed into another set, S_{32}. Similarly, there are no cargo routes where all flights belong to the second-leg set. Consequently, these two sets also satisfy the Ghouila-Houri characterization.

For the Type 4 matrix, only the first-leg set and the last-leg set are present. If there is a cargo route with a flight that can be categorized into the second-leg set, that route must be classified as Type 3. Therefore, we assign the rows corresponding to the first leg to set S_{41} while the remaining rows are placed in set S_{42}. It is evident that these two sets also satisfy the Ghouila-Houri characterization.

Under the operational time available for the cargo aircraft, there are three flights that make up an aircraft route at most. If connections of cargo routes in Type 2 are all through cargo connections with no transshipment at hub airports, then cargo routes in Type 2 and Type 4 would not affect each other. That is, given the values of f_k^l and x_k^p, if a cargo route in Type 2 is $l_1 - l_2 - l_3$ while the origin and destination airports of l_2 are non-hub airports, then there would not be a flight $l_j \in L$ that can constitute a cargo route as $l_1 - l_j$ or $l_j - l_3$ in Type 4. Then we can easily put the rows in S_{21},S_{31} and S_{41} into a set S_1, the other rows in the constraint matrix are put into another aet S_2. Those two sets satisfy the Ghouila-Houri characterization.

Considering the case in which flights in S_{21} can construct a Type 4 cargo route, suppose there are two cargo routes, $l_1 - l_2 - l_3$ and $l_4 - l_5 - l_6$, classified as Type 2. In this case, l_1 and l_6 can form the route $l_1 - l_6$ within Type 4, indicating that both the destination airport of l_1 and the origin airport of l_6 are hub airports. Otherwise, the route $l_1 - l_6$ would not exist, as standard cargo transshipment can only occur at the hub airports. In this context, the flights l_1, l_3 and l_5 can be assigned to set S_1, while l_2, l_4 and l_6 are assigned to set S_2. Consequently, within our network, we can categorize the rows in the Type 2 matrix into two disjoint sets, ensuring that no flights can be connected within each set. Consequently, all rows in Type 3, along with the affected rows in Types 1 and 4, would be included in these two sets.

In conclusion, given the values of f_k^l and x_k^p, the relevant matrices of constraints all conform to the Ghouila-Houri characterization. This implies that the aforementioned constraint coefficient matrix is a totally unimodular matrix. □

Appendix C. Benders-based method

This section details the application of the Benders Decomposition (BD) algorithm to solve the problem. The BD algorithm partitions the original model into a Benders master problem (BMP) and a Benders subproblem (BSP). The BD algorithm proceeds iteratively: first, the BMP is solved to obtain a candidate solution. This solution is then passed to the BSP, which is subsequently solved to generate either a feasibility cut or an optimality cut. This generated cut is then added into the BMP, which is solved again in the next iteration. Using the solutions from the BMP and BSP, we can update the lower bounds and upper bounds of the objective value of the original problem. The BD algorithm terminates when the gap between the lower bound and upper bound reaches a pre-set threshold $\epsilon = 0.05\%$, or the overall time limit is reached. The BMP is formulated as follows:

$$[\text{BMP}]\ \min \quad \sum_{k\in K}\sum_{l\in L} c_k^l f_k^l + \theta \tag{C.1}$$

$$\text{s.t.} \quad \sum_{k\in K} f_k^l \le 1, \qquad \forall l \in L \tag{C.2}$$

$$\sum_{p\in P} x_k^p \le N_k, \qquad \forall k \in K \tag{C.3}$$

$$\sum_{o^p=i,p\in P} x_k^p - \sum_{d^p=i,p\in P} x_k^p = 0, \qquad \forall i \in H, k \in K \tag{C.4}$$

$$\sum_{p\in P_k^l} x_k^p = f_k^l, \qquad \forall l \in L, k \in K \tag{C.5}$$

$$\text{feasibility cuts}, \tag{C.6}$$

$$\text{optimality cuts}, \tag{C.7}$$

$$x_k^p \in \{0,1\}, \qquad \forall p \in P, k \in K \tag{C.8}$$

$$f_k^l \in \{0,1\}, \qquad \forall l \in L, k \in K \tag{C.9}$$

$$\theta \ge 0, \tag{C.10}$$

where θ is a surrogate variable representing the objective value of the BSP. Due to the computational difficulty of solving the BMP, its solution process at each iteration is terminated once the optimality gap reaches 10%. Given a solution $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{f}})$ from the BMP, the BSP is formulated based on the TUM property as follows:

$$[\text{BSP}]\ \min \quad \sum_{q\in Q}\sum_{r\in R} u_r^q y_r^q + \sum_{q\in Q}\sum_{b\in B} \beta_b^q \varphi_b^q + \sum_{q\in Q} \eta^q z^q \tag{C.11}$$

$$\text{s.t.} \quad \sum_{r\in R} y_r^q + \sum_{b\in B_q} \varphi_b^q + z^q = w^q, \qquad \forall q \in Q \quad (\zeta_q) \tag{C.12}$$

$$\sum_{q\in Q}\sum_{r\in R_k^l} y_r^q \le v^k \bar{f}_k^l, \qquad \forall l \in L, k \in K \quad (\xi_k^l) \tag{C.13}$$

$$\sum_{q\in Q_b} \varphi_b^q \le \varpi_b, \qquad \forall b \in B \quad (\chi_b) \tag{C.14}$$

$$\gamma_{(r,l_m,l_n,k)}^q - y_r^q = 0, \qquad \forall (r,l_m,l_n,k) \in TC, q \in Q_k^{l_m} \cup Q_k^{l_n} \quad (\varrho_{(r,l_m,l_n,k)}^q) \tag{C.15}$$

$$\gamma_{(r,l_m,l_n,k)}^q \le \min\{w^q, v_k\} \sum_{p\in P_{(l_m,l_n)}^k} \bar{x}_k^p, \qquad \forall (r,l_m,l_n,k) \in TC, q \in Q_k^{l_m} \cup Q_k^{l_n} \quad (\vartheta_{(r,l_m,l_n,k)}^q) \tag{C.16}$$

$$y_r^q \geq 0, \qquad \forall q \in Q, r \in R \tag{C.17}$$
$$\varphi_b^q \geq 0, \qquad \forall q \in Q, b \in B \tag{C.18}$$
$$z^q \geq 0, \qquad \forall q \in Q \tag{C.19}$$
$$\gamma^q_{(r,l_m,l_n,k)} \geq 0, \qquad \forall (r,l_m,l_n,k) \in TC, q \in Q_k^{l_m} \cup Q_k^{l_n}. \tag{C.20}$$

Let $\zeta_q, \xi_k^l, \chi_b, \varrho^q_{(r,l_m,l_n,k)}, \vartheta^q_{(r,l_m,l_n,k)}$ be the dual variables associated with constraints (C.12)–(C.16), respectively. For notational convenience and mathematical consistency, we extend the domain of the variable $\varrho^q_{(r,l_m,l_n,k)}$ from the subset $Q_k^{l_m} \cup Q_k^{l_n}$ to the entire demand set Q by setting $\varrho^q_{(r,l_m,l_n,k)} = 0$ for all $(r,l_m,l_n,k) \in TC$ and $q \in Q \setminus (Q_k^{l_m} \cup Q_k^{l_n})$. The dual problem of the BSP is formulated as follows:

$$\begin{aligned}\max \quad & \sum_{q\in Q} w^q \zeta_q + \sum_{l\in L}\sum_{k\in K}(v^k \bar{f}_k^l)\xi_k^l + \sum_{b\in B}\varpi_b \chi_b \\ & + \sum_{(r,l_m,l_n,k)\in TC}\sum_{q\in Q_k^{l_m}\cup Q_k^{l_n}}\left(\min\{w^q, v_k\}\sum_{p\in P^k_{(l_m,l_n)}}\bar{x}_k^p\right)\vartheta^q_{(r,l_m,l_n,k)}\end{aligned} \tag{C.21}$$
$$\text{s.t.} \quad \zeta_q + \sum_{l\in L}\sum_{k\in K} g^r_{(l,k)}\xi_k^l - \sum_{(r,l_m,l_n,k)\in TC_r}\varrho^q_{(r,l_m,l_n,k)} \leq u_r^q, \qquad \forall q \in Q, r \in R \tag{C.22}$$
$$\zeta_q + \chi_b \leq \beta_b^q, \qquad \forall q \in Q, b \in B_q \tag{C.23}$$
$$\zeta_q \leq \eta^q, \qquad \forall q \in Q \tag{C.24}$$
$$\varrho^q_{(r,l_m,l_n,k)} + \vartheta^q_{(r,l_m,l_n,k)} \leq 0, \qquad \forall (r,l_m,l_n,k) \in TC, q \in Q_k^{l_m} \cup Q_k^{l_n} \tag{C.25}$$
$$\xi_k^l \leq 0, \forall l \in L, k \in K \tag{C.26}$$
$$\chi_b \leq 0, \qquad \forall b \in B \tag{C.27}$$
$$\vartheta^q_{(r,l_m,l_n,k)} \leq 0, \qquad \forall (r,l_m,l_n,k) \in TC, q \in Q_k^{l_m} \cup Q_k^{l_n} \tag{C.28}$$
$$\varrho^q_{(r,l_m,l_n,k)} = 0, \qquad \forall (r,l_m,l_n,k) \in TC, q \in Q \setminus (Q_k^{l_m} \cup Q_k^{l_n}), \tag{C.29}$$

where $g^r_{(l,k)} = 1$ if a cargo route r is valid for flight slot l and aircraft type k (i.e., $r \in R_k^l$), and $g^r_{(l,k)} = 0$ otherwise. And TC_r is a subset of TC related to cargo route r, which may be empty.

Because for each cargo $q \in Q$, we can construct a feasible solution for the BSP by setting $z^q = w^q$, $y_r^q = 0\ \forall r \in R$, and $\varphi_b^q = 0\ \forall b \in B$. Under this setting, all the constraints in the BSP are satisfied automatically, and the objective value of the BSP is bounded to $\sum_{q\in Q}\eta^q w^q$. Thus, the BSP is always feasible and bounded. According to strong duality, the dual problem of the BSP is always feasible and bounded. Hence, we only need to add optimality cuts to the BMP. Let Δ denote the polyhedron defined by constraints (C.22)–(C.29), and let Γ_Δ be the set of extreme points of Δ. The optimality cut added to the BMP can be expressed as follows:

$$\theta \geq \sum_{q\in Q} w^q\zeta_q + \sum_{l\in L}\sum_{k\in K}(u^k f_k^l)\xi_k^l + \sum_{b\in B}\varpi_b\chi_b + \sum_{(r,l_m,l_n,k)\in TC}\sum_{q\in Q_k^{l_m}\cup Q_k^{l_n}}\left(\min\{w^q, v_k\}\sum_{p\in P^k_{(l_m,l_n)}}x_k^p\right)\vartheta^q_{(r,l_m,l_n,k)},$$
$$\forall (\boldsymbol{\zeta}, \boldsymbol{\xi}, \boldsymbol{\chi}, \boldsymbol{\varrho}, \boldsymbol{\vartheta}) \in \Gamma_\Delta$$

Appendix D. Benchmark performance results

Appendix E. Model variants

The three models are primarily distinguished by their methods for generating cargo transshipment routes utilizing cargo flights, while ensuring consistency in operational constraints related to the cargo fleet and belly capacity. In contrast to **Model TN**, **Model O** exclusively utilizes standard transshipment connections for generating cargo transshipment routes, thus excluding through cargo connections. As a result, constraints (3i) and (3j), which pertain to through cargo connections, are not applicable and are omitted from this model. Let R_1 denote the set of feasible cargo routes using cargo flights in **Model O**. The formulation of **Model O** is as follows.

$$\min \quad \sum_{q\in Q}\sum_{r\in R_1} u_r^q y_r^q + \sum_{q\in Q}\sum_{b\in B}\beta_b\varphi_b^q + \sum_{q\in Q}\eta^q z^q + \sum_{k\in K}\sum_{l\in L} c_k^l f_k^l$$

s.t.

$$(3b)-(3e), (3h), (3k)-(3l), (3o)-(3p)$$
$$\sum_{r\in R_1} y_r^q + \sum_{b\in B_q}\varphi_b^q + z^q = w^q, \qquad \forall q \in Q$$

Table D.11
Experimental results comparison of small-scale instances.

Instance	CPLEX		CG					Benders			
	Obj	time(s)	Obj	time(s)	CG.time(s)	CG.Num	Ip.gap	Obj	time(s)	Iterations	Ip.gap
1	3077508.64	9.58	3077508.64	5.99	3.04	51	0.00%	7012398.82	TL	17	127.86%
2	3049679.42	9.86	3049679.42	8.37	3.56	54	0.00%	7245307.39	TL	16	137.58%
3	3209616.42	16.16	3209616.42	14.11	5.48	58	0.00%	7050932.77	TL	18	119.68%
4	3237875.09	16.67	3237875.09	10.64	5.18	55	0.00%	6831266.80	TL	15	110.98%
5	3494104.51	25.42	3494104.51	19.50	4.39	74	0.00%	7050932.77	TL	21	101.80%
6	3258069.05	27.92	3258069.05	23.21	12.60	46	0.00%	8336005.27	TL	21	155.86%
7	4255750.67	33.05	4255750.67	23.67	10.68	64	0.00%	7378056.61	TL	25	73.37%
8	4571077.40	35.07	4571077.40	15.39	3.71	63	0.00%	8038705.68	TL	22	75.86%
9	4092144.21	37.82	4092144.21	26.38	14.52	57	0.00%	7984090.61	TL	23	95.11%
10	5157459.50	51.13	5157459.50	10.86	2.88	53	0.00%	8653251.76	TL	21	67.78%
11	4316882.68	57.61	4316882.68	29.47	14.51	53	0.00%	7615950.97	TL	23	76.42%
12	3644077.18	69.29	3644077.18	23.97	9.81	49	0.00%	8940601.39	TL	17	145.35%
13	4899055.76	80.07	4899055.76	59.48	20.00	72	0.00%	8859707.98	TL	24	80.85%
14	4798325.10	90.40	4798325.10	35.40	12.49	76	0.00%	8759872.25	TL	22	82.56%
15	4532209.28	92.12	4534659.32	27.20	8.39	52	0.05%	8508956.32	TL	23	87.74%
16	4472921.05	127.84	4472921.05	12.91	3.01	51	0.00%	8675510.50	TL	25	93.96%
17	4721504.55	135.96	4721504.55	44.59	15.35	57	0.00%	7858074.72	TL	23	66.43%
18	4966981.28	140.22	4966981.28	17.57	3.02	51	0.00%	8590859.51	TL	23	72.96%
19	3418750.07	173.25	3437675.92	28.11	6.08	52	0.55%	8915220.48	TL	19	160.77%
20	4043901.03	180.69	4043901.03	30.63	3.48	57	0.00%	7508818.95	TL	24	85.68%
21	4340854.27	201.24	4340854.27	90.30	18.50	65	0.00%	7915425.44	TL	22	82.35%
22	3185639.59	223.76	3199540.01	44.85	11.65	56	0.44%	9211239.79	TL	21	189.15%
23	4101111.97	265.66	4101843.82	55.67	6.54	60	0.02%	8947220.70	TL	18	118.17%
24	4648230.43	361.79	4648230.43	153.67	15.29	60	0.00%	9346680.30	TL	23	101.08%
25	6193499.27	507.81	6193499.27	64.83	10.72	68	0.00%	8483412.39	TL	24	36.97%
26	3965332.87	527.11	3965332.87	46.58	8.57	62	0.00%	8699714.22	TL	23	119.39%
27	5959763.00	737.57	5959763.00	155.59	13.40	68	0.00%	8639325.32	TL	29	44.96%
28	3618362.37	793.35	3630794.49	152.50	6.72	56	0.34%	8105682.70	TL	23	124.02%
29	4669026.45	1115.70	4669026.45	126.05	17.02	65	0.00%	8405757.98	TL	24	80.03%
30	5403163.99	1231.27	5406597.55	185.11	5.48	56	0.06%	8204991.43	TL	21	51.86%
31	6246092.07	1314.24	6246092.08	75.10	20.30	73	0.00%	9043084.34	TL	23	44.78%
32	5242828.93	1537.94	5242828.93	114.07	8.30	51	0.00%	8108617.68	TL	21	54.66%
33	6621099.63	1647.74	6621099.63	59.21	14.79	58	0.00%	9207745.83	TL	20	39.07%
34	5406091.74	1898.81	5411234.58	103.77	10.78	53	0.10%	7895365.08	TL	23	46.05%
35	5616535.07	TL	5616535.07	238.54	19.12	71	0.00%	9620509.31	TL	24	71.29%
36	3887395.95	TL	3887395.95	278.01	16.42	59	0.00%	9215034.38	TL	22	137.05%
37	3322639.22	TL	3322639.22	331.05	6.87	46	0.00%	7049795.11	TL	22	112.17%
38	4093133.14	TL	4086834.34	507.74	8.55	54	-0.15%	7399055.05	TL	20	80.77%
39	3112151.71	TL	3112151.71	616.80	10.39	49	0.00%	7208009.07	TL	17	131.61%
40	4950406.35	TL	4950406.35	688.48	16.30	60	0.00%	9663087.11	TL	26	95.20%
41	6074768.56	TL	6074768.56	762.69	14.58	69	0.00%	8782325.34	TL	23	44.57%
42	4960279.33	TL	4960279.33	824.94	21.49	81	0.00%	8859362.42	TL	22	78.61%
43	2819569.95	TL	2827705.32	848.49	11.63	49	0.29%	7196823.35	TL	15	155.25%
44	4159112.30	TL	4132371.33	1160.90	7.86	54	-0.64%	6714651.56	TL	22	61.44%
45	4302556.59	TL	4316547.95	1462.26	6.15	56	0.33%	9134569.67	TL	18	112.31%
46	3787689.82	TL	3787689.82	1586.12	12.04	83	0.00%	7894223.84	TL	20	108.42%
47	3988652.44	TL	3988652.44	1711.47	14.62	96	0.00%	6432991.45	TL	22	61.28%
48	3970860.18	TL	3970860.18	1762.81	14.75	103	0.00%	7392490.41	TL	19	86.17%
49	3916057.58	TL	3916057.58	1940.19	10.85	46	0.00%	6565300.68	TL	24	67.65%
50	3834294.93	TL	3834294.93	TL	7.76	50	0.00%	7387491.32	TL	21	92.67%
Average	—	1428.14	—	404.34	10.59	60.44	0.03%	—	3600.00	21.48	92.95%

$$\sum_{q\in Q}\sum_{r\in R_k^l\cap R_1} y_r^q \le \upsilon^k f_k^l, \qquad \forall l\in L, k\in K \tag{E.1}$$

$$y_r^q \in \mathbb{N}, \qquad \forall q\in Q, r\in R_1$$

Model TH incorporates restricted through cargo connections into cargo transshipment routes, allowing such connections exclusively at hub airports. Let R_2 denote the set of feasible cargo routes using cargo flights in **Model TH**. Consequently, we have $R_1 \subseteq R_2 \subseteq R$, where R represents the set of cargo routes using cargo flights in **Model TN**. We also let $TC_2 \subseteq TC$ denote the set of through cargo connections in **Model TH**. The formulation of **Model TH** is as follows.

$$\min \sum_{q\in Q}\sum_{r\in R_2} u_r^q y_r^q + \sum_{q\in Q}\sum_{b\in B}\beta_b\varphi_b^q + \sum_{q\in Q}\eta^q z^q + \sum_{k\in K}\sum_{l\in L} c_k^l f_k^l$$

s.t.

$$
\begin{aligned}
&(3b)-(3e),(3h),(3k)-(3l),(3o)-(3p)\\
&\sum_{r\in R_2} y_r^q + \sum_{b\in B_q} \varphi_b^q + z^q = w^q, \qquad \forall q \in Q\\
&\sum_{q\in Q}\sum_{r\in R_k^l\cap R_2} y_r^q \le v^k f_k^l, \qquad \forall l \in L, k \in K\\
&\gamma^q_{(r,l_m,l_n,k)} = y_r^q, \qquad \forall (r,l_m,l_n,k) \in TC_2, q \in Q_k^{l_m} \cup Q_k^{l_n}\\
&\gamma^q_{(r,l_m,l_n,k)} \le \min\{w^q, v_k\} \sum_{p\in P^k_{(l_m,l_n)}} x_k^p, \qquad \forall (r,l_m,l_n,k) \in TC_2, q \in Q_k^{l_m} \cup Q_k^{l_n}\\
&\gamma^q_{(r,l_m,l_n,k)} \in \mathbb{N}, \qquad \forall (r,l_m,l_n,k) \in TC_2, q \in Q_k^{l_m} \cup Q_k^{l_n}\\
&y_r^q \in \mathbb{N}, \qquad \forall q \in Q, r \in R_2
\end{aligned}
\tag{E.2}
$$

References

Abara, J., 1989. Applying integer linear programming to the fleet assignment problem. Interfaces (Providence) 19 (4), 20–28.

Al-Thani, N.A., Ahmed, M.B., Haouari, M., 2016. A model and optimization-based heuristic for the operational aircraft maintenance routing problem. Transp. Res. Part C Emerg. Technol. 72, 29–44.

Armacost, A.P., Barnhart, C., Ware, K.A., 2002. Composite variable formulations for express shipment service network design. Transp. Sci. 36 (1), 1–20.

Azadian, F., Murat, A.E., Chinnam, R.B., 2012. Dynamic routing of time-sensitive air cargo using real-time information. Transp. Res. Part E Logist. Transp. Rev. 48 (1), 355–372.

Bang-Jensen, J., Gutin, G.Z., 2008. Digraphs: theory, algorithms and applications. Springer Science & Business Media.

Barnhart, C., Boland, N.L., Clarke, L.W., Johnson, E.L., Nemhauser, G.L., Shenoi, R.G., 1998. Flight string models for aircraft fleeting and routing. Transp. Sci. 32 (3), 208–220.

Barnhart, C., Cohn, A., 2004. Airline schedule planning: accomplishments and opportunities. Manuf. Serv. Oper. Manag. 6 (1), 3–22.

Bellman, R., 1962. Dynamic programming treatment of the travelling salesman problem. J. ACM (JACM) 9 (1), 61–63.

Berge, M.E., Hopperstad, C.A., 1993. Demand driven dispatch: a method for dynamic aircraft capacity assignment, models and algorithms. Oper. Res. 41 (1), 153–168.

Birolini, S., Jacquillat, A., 2023. Day-ahead aircraft routing with data-driven primary delay predictions. Eur. J. Oper. Res. 310 (1), 379–396.

Boeing, 2020. World air cargo forecast 2020–2039. https://www.boeing.com/content/dam/boeing/boeingdotcom/market/assets/downloads/2020_WACF_PDF_Download.pdf.

Bulbul, K.G., Kasimbeyli, R., 2021. Augmented lagrangian based hybrid subgradient method for solving aircraft maintenance routing problem. Comput. Operat. Res. 132, 105294.

Christofides, N., 1975. Graph theory: An algorithmic approach (Computer science and applied mathematics). Academic Press, Inc.

Derigs, U., Friederichs, S., 2013. Air cargo scheduling: integrated models and solution procedures. OR Spectrum 35 (2), 325–362.

Derigs, U., Friederichs, S., Schäfer, S., 2009. A new approach for air cargo network planning. Transp. Sci. 43 (3), 370–380.

Desaulniers, G., Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F., 1997. Daily aircraft routing and scheduling. Manage. Sci. 43 (6), 841–855.

Etschmaier, M.M., Mathaisel, D. F.X., 1985. Airline scheduling: an overview. Transp. Sci. 19 (2), 127–138.

Ghouila-Houri, A., 1964. Flots et tensions dans un graphe. In: Annales Scientifiques de l'École Normale Supérieure. Vol. 81, pp. 267–339.

Huang, L., Xiao, F., Zhou, J., Duan, Z., Zhang, H., Liang, Z., 2023. A machine learning based column-and-row generation approach for integrated air cargo recovery problem. Transp. Res. Part B: Methodol. 178, 102846.

IATA, 2020. The e-commerce impact on air cargo operations. https://www.iata.org/contentassets/d22340c37e0c4cfd8fc05ca6ebf6cc9f/ecommerce-impact-challenges.pdf/.

Jamili, A., 2017. A robust mathematical model and heuristic algorithms for integrated aircraft routing and scheduling, with consideration of fleet assignment problem. J. Air Transp. Manag. 58, 21–30.

Kenan, N., Jebali, A., Diabat, A., 2018. The integrated aircraft routing problem with optional flights and delay considerations. Transp. Res. Part E: Logist. Transp. Rev. 118, 355–375.

Khaled, O., Minoux, M., Mousseau, V., Michel, S., Ceugniet, X., 2018. A compact optimization model for the tail assignment problem. Eur. J. Oper. Res. 264 (2), 548–557.

Lee, C. K.M., Zhang, S., Ng, K. K.H., 2019. Design of an integration model for air cargo transportation network design and flight route selection. Sustainability 11 (19), 5197.

Li, D., Huang, H.-C., Chew, E.-P., Morton, A.D., 2007. Simultaneous fleet assignment and cargo routing using benders decomposition. Container Termin. Cargo Syst.: Design, Operations Manag., Logist. Control Issues , 315–331.

Li, Z., Bookbinder, J.H., Elhedhli, S., 2012. Optimal shipment decisions for an airfreight forwarder: formulation and solution methods. Transp. Res. Part C: Emerg. Technol. 21 (1), 17–30.

Liu, Y., Yin, M., Hansen, M., 2019. Economic costs of air cargo flight delays related to late package deliveries. Transp. Res. Part E: Logist. Transp. Rev. 125, 388–401.

Lohatepanont, M., Barnhart, C., 2004. Airline schedule planning: integrated models and algorithms for schedule design and fleet assignment. Transp. Sci. 38 (1), 19–32.

Molenbruch, Y., Braekers, K., Eisenhandler, O., Kaspi, M., 2023. The electric dial-a-ride problem on a fixed circuit. Transp. Sci. 57 (3), 594–612.

Papadakos, N., 2009. Integrated airline scheduling. Comput. Operat. Res. 36 (1), 176–195. Part Special Issue: Operations Research Approaches for Disaster Recovery Planning.

Roelen, A. L.C., Pikaar, A.J., Ovaa, W., 2000. An analysis of the safety performance of air cargo operators. Technical Report NLRTP-2000-210. National Aerospace Laboratory NLR. Amsterdam.

Safaei, N., Jardine, A. K.S., 2018. Aircraft routing with generalized maintenance constraints. Omega (Westport) 80, 111–122.

Sherali, H.D., Bae, K.-H., Haouari, M., 2013. An integrated approach for airline flight selection and timing, fleet assignment, and aircraft routing. Transp. Sci. 47 (4), 455–476.

Sherali, H.D., Bish, E.K., Zhu, X., 2006. Airline fleet assignment concepts, models, and algorithms. Eur. J. Oper. Res. 172 (1), 1–30.

Tang, C.-H., Yan, S., Chen, Y.-H., 2008. An integrated model and solution algorithms for passenger, cargo, and combi flight scheduling. Transp. Res. Part E: Logist. Transp. Rev. 44 (6), 1004–1024.

Wei, K., Vaze, V., Jacquillat, A., 2020. Airline timetable development and fleet assignment incorporating passenger choice. Transp. Sci. 54 (1), 139–163.

Wen, X., Sun, X., Ma, H.-L., Sun, Y., 2022. A column generation approach for operational flight scheduling and aircraft maintenance routing. J. Air Transp. Manag. 105, 102270.

Xiao, F., Guo, S., Huang, L., Huang, L., Liang, Z., 2022. Integrated aircraft tail assignment and cargo routing problem with through cargo consideration. Transp. Res. Part B: Methodol. 162, 328–351.

Xu, Y., Wandelt, S., Sun, X., 2021. Airline integrated robust scheduling with a variable neighborhood search based heuristic. Transp. Res. Part B: Methodol. 149, 181–203.

Yan, S., Chen, S.-C., Chen, C.-H., 2006. Air cargo fleet routing and timetable setting with multiple on-time demands. Transp. Res. Part E: Logist. Transp. Rev. 42 (5), 409–430.

Yan, S., Tang, C.-H., Fu, T.-C., 2008. An airline scheduling model and solution algorithms under stochastic demands. Eur. J. Oper. Res. 190 (1), 22–39.

Yan, S., Tang, C.-H., Lee, M.-C., 2007. A flight scheduling model for taiwan airlines under market competitions. Omega (Westport) 35 (1), 61–74.

Yan, S., Tseng, C.-H., 2002. A passenger demand model for airline flight scheduling and fleet routing. Comput. Operat. Res. 29 (11), 1559–1581.

Yan, S., Young, H.-F., 1996. A decision support framework for multi-fleet routing and multi-stop flight scheduling. Transp. Res. Part A: Policy Pract. 30 (5), 379–398.

Yıldız, B., Savelsbergh, M., 2022. Optimizing package express operations in china. Eur. J. Oper. Res. 300 (1), 320–335.

Yu, S., Jiang, Y., 2024. Network design and delivery scheme optimisation under integrated air-rail freight transportation. Int. J. Logist. Res. Appl. 27 (3), 411–427.

Yu, S., Yang, Z., Yu, B., 2017. Air express network design based on express path choices–chinese case study. J. Air Transp. Manage. 61, 73–80.

Zheng, H., Sun, H., Zhu, S., Kang, L., Wu, J., 2023. Air cargo network planning and scheduling problem with minimum stay time: a matrix-based ALNS heuristic. Transp. Res. Part C: Emerg. Technol. 156, 104307.

Zhou, L., Liang, Z., Chou, C.-A., Chaovalitwongse, W.A., 2020. Airline planning and scheduling: models and solution methodologies. Front. Eng. Manag. 7 (1), 1–26.