JOINT OPTIMIZATION OF AIRPORT SLOT ALLOCATION AND GATE ASSIGNMENT

XUANJI LONG

MPhil

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University

Department of Aeronautical and Aviation Engineering

Joint Optimization of Airport Slot Allocation and Gate Assignment

Xuanji LONG

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Philosophy

August 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

__________________________(Signed)

Xuanji Long (Name of student)

Abstract

During peak operating periods, airports must coordinate flight schedules and gate usage under intertwined risks of capacity congestion, knock-on delays, and limited gate availability. This study investigates the joint optimization of airport slot allocation and gate assignment under capacity constraints with operational uncertainty (e.g., flight delays). Integrating slot allocation with gate assignment is proposed to mitigate inefficiencies from decoupled decisions and the deadweight losses arising from downstream gate conflicts. To better reflect real-world operations, we formulate gate assignment as a real-time problem in which uncertainty is progressively revealed over time. We show that jointly optimizing slots and gates enables temporal smoothing of traffic peaks and alleviates turnaround conflicts. The analysis is then extended to large-scale, realistic settings with multiple airlines and terminals.

We develop two optimization schemes to realize joint slot allocation and gate assignment decisions under uncertainty. First, a two-stage stochastic integer program treats gate assignment as a rolling, real-time recourse decision that incorporates updated delay information; a Benders-based decomposition combined with a rolling horizon approach and tabu search subproblems yields tractable solutions. Second, we enhance slot allocation resilience with a two-stage robust optimization model, solved using a nested column-and-constraint generation algorithm. Using extensive historical data from Shanghai Hongqiao International Airport, we demonstrate that both schemes solve large instances efficiently and achieve significant gains in computational efficiency and operational performance. These results support the practical applicability of joint optimization of slot allocation and gate assignment in airport scheduling and operations for uncertainty-aware capacity management.

Publications arising from the thesis

Journal article:

Long, X., Lyu, M & Liu, W*. (2025). Joint Optimization of Airport Slot Allocation and Gate Assignment: A Stochastic Integer Programming Approach, *Computers & Industrial Engineering*. (Under 2nd review)

Conference article:

Long, X., & Liu, W*. (2025, Jul 1-4) A Two-Stage Robust Optimization for Airport Slot Allocation and Gate Assignment [Paper Presentation]. *The 28th Air Transport Research Society (ATRS) World Conference*, Hong Kong

Long, X., Lyu, M & Liu, W*. (2024, Dec 9-10). Joint Optimization of Airport Slot Allocation and Gate Assignment: A Stochastic Integer Programming Approach [Paper Presentation]. *Proceedings of the 28th International Conference of Hong Kong Society for Transportation Studies (HKSTS 2024)*, Hong Kong

Acknowledgements

I extend my deepest gratitude to my chief supervisor, Prof. Wei Liu, for his steadfast guidance, support, and encouragement throughout this research journey. I am sincerely thankful to Prof. Feng Zhu from Nanyang Technological University and Prof. Yong-Hong Kuo from the University of Hong Kong for serving as the examiners of my thesis. Finally, I sincerely thank my parents, colleagues, and friends for their unwavering support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The global aviation industry operates on a delicate balance between growing demand and constrained infrastructure. Globally, over 200 International Air Transport Association (IATA) Level 3 airports continue to experience chronic demand-capacity imbalances. These airports are not just regional hubs but critical nodes in the global network, handling 40% of worldwide passenger traffic and over 55% of aircraft movements outside the U.S. The post-COVID-19 surge in air travel has intensified this strain, with passenger numbers in 2023 exceeding pre-pandemic levels by 15% in Asia-Pacific markets (ACI, 2024). This recovery, while economically vital, has reignited systemic inefficiencies: flight delays, gate overcrowding, and cascading disruptions that ripple across networks. In China, for instance, 13.2% of flights experienced delays in 2023, costing airlines an estimated $2.7 billion annually in direct operational costs and lost revenue (CAAC, 2024).

Airport slot allocation (ASA), an administrative scheduling process, is the primary mechanism for managing demand at congested airports. Yet, traditional ASA frameworks operate in a vacuum, prioritizing strategic scheduling while ignoring operational realities. A glaring example is the disconnect between ASA and gate assignment. Gates are a finite resource that directly impacts an airport's operational capacity: a delayed flight due to weather (60% of disruptions in China) or airline operations (15%) forces real-time gate reassignments, disrupting downstream schedules and amplifying delays. This siloed approach leaves airports trapped in a cycle of reactive adjustments, where meticulously crafted schedules unravel under routine uncertainties.

While infrastructure expansion remains the ultimate solution, its feasibility is limited. Major airports like London Heathrow and Tokyo Haneda face decades-long delays in runway projects due to environmental litigation and community opposition. Market-based mechanisms, such as congestion pricing, face regulatory hurdles and airline resistance. Administrative ASA, though widely adopted, is hamstrung by its rigidity. Following the World Airport Slot Guidelines (WASG), many models place historical rights ("grandfather rights") and priority rules above operational adaptability. For example, a 2022 study of European hubs found that 25% of allocated slots went unused due to airlines overbidding for peak times, exacerbating congestion without improving utilization (Jorge et al., 2021). These limitations underscore a critical gap: current ASA frameworks lack mechanisms to align strategic plans with operational resilience.

The interdependence between ASA and gate assignment presents both a challenge and an opportunity. Slot directly impact gate availability, while gate constraints—such as size compatibility and turnaround times—can retroactively restrict slot feasibility. For example, a wide-body aircraft requiring a specialized gate cannot be allocated a slot during peak hours if all compatible gates are occupied. Traditional ASA models often neglect these interdependencies, resulting in schedules that may be theoretically feasible but operationally fragile.

This thesis is driven by a transformative vision: integrating ASA with gate assignment under uncertainty to develop adaptive and robust scheduling frameworks. By bridging strategic planning with real-time operations, airports can proactively mitigate disruptions. For instance, a robust flight schedule might incorporate buffer gates for high-priority flights under worst-case scenarios, while a stochastic model could optimize slot allocation by minimizing expected delays based on historical data. This integrated approach aligns with emerging industry trends, such as IATA's 2023 revision of the WASG, which emphasizes collaborative decision-making and operational flexibility.

1.2 Airport slot allocation and gate assignment

This section outlines the processes of airport slot allocation and gate assignment and discusses their interdependence.

1.2.1 Airport slot allocation process

At capacity-constrained airports, airline access is governed through ASA under the WASG. This global framework, applied seasonally, covers roughly 200 airports worldwide.

(1) Pre-Season activities

Before each scheduling season, the coordinator publishes declared capacities and scheduling parameters that bound feasible demand:

1. Declared runway, terminal, and apron capacities by time-of-day, including any hour- or period-specific limits.

2. Coordination parameters, such as minimum turnaround times, aircraft size constraints, night curfews, and noise restrictions.

3. Slot pool definitions and any allocation priorities mandated by regulation.

Airlines then submit initial schedules by the industry timetable milestone (e.g., IATA Slot Conference deadlines). Submissions include flight numbers, aircraft types, requested arrival/departure times, turnaround plans, and historic rights claims. The coordinator validates requests against the declared parameters, identifies over-subscribed periods, and prepares for iterative coordination at the Slot Conference.

(2) Initial slot allocation

After pre-season submissions, the coordinator issues the first season schedule by applying WASG priority rules under declared capacity limits. Historic slots that satisfy the "use it or lose it" rule are confirmed with minimal timing changes, while returned capacity and newly available slots form a pool allocated with priority to new entrants and remedy obligations. Requests are checked against runway, terminal, apron, and operational constraints (curfews, noise, turnaround). Where peak periods are oversubscribed, the coordinator proposes nearby times or swaps to smooth demand and ensure feasible turns. Results are communicated via Initial Coordination Messages, explaining any deviations and inviting airlines to respond ahead of the Slot Conference.

(3) Post-allocation activities

Following the initial coordination, airlines and the coordinator refine the schedule through an iterative process leading up to the Slot Conference. Carriers review initial coordination messages, accept or reject offers, and submit revised preferences within published time windows. The coordinator reconciles responses, resolves residual conflicts, and re-issues updates while maintaining adherence to declared runway, terminal, and apron capacities, as well as curfews and noise limits.

Schedule optimization continues as airlines seek better times, align connections, and adjust aircraft types. Changes generate a dynamic slot pool as slots are returned, exchanged, or re-timed, subject to rules on series integrity and minimum usage. Where necessary, the coordinator facilitates bilateral swaps and multi-party trades to reduce peaks and preserve viable turns, ensuring that arrival and departure slots remain operable as a pair.

Compliance and transparency are maintained through audit trails and publication of coordination parameters and rationales for decisions. Historic precedence is monitored against the applicable usage threshold, with underused series flagged for potential retraction. The post-allocation phase concludes at the Slot Conference, where face-to-face negotiation consolidates outstanding adjustments, after which a coordinated schedule is issued for the season and ongoing in-season management begins.

1.2.2 Gate assignment

As a critical operational decision-making process at airports, gate assignment significantly influences airport efficiency, passenger satisfaction, and resource utilization (Evler et al., 2021). This process involves assigning available gates to arriving and departing flights while adhering to a range of physical constraints, such as gate size compatibility, aircraft type, and turnaround times. Additionally, operational requirements, including minimizing passenger walking distances, avoiding gate conflicts, and ensuring smooth connections for transfer passengers, must be considered. The quality of gate assignment underpins turnaround efficiency and customer experience, rendering it essential to airport operations.

The complexity of gate assignment stems from its highly dynamic nature. Flight schedules are prone to frequent disruptions caused by various uncertainties, such as weather delays, mechanical malfunctions, air traffic congestion, and crew availability. These disruptions necessitate rapid and effective adjustments to gate assignments in real-time, further complicating the problem. Inefficient gate allocation in such scenarios can lead to cascading delays, increased passenger dissatisfaction, and suboptimal utilization of critical airport resources.

Traditionally, gate assignment has been approached as a short-term, deterministic optimization problem. These conventional approaches focus on tactical objectives, such as minimizing passenger transit times, reducing aircraft taxiing distances, or maximizing gate utilization during peak operational periods. However, these methods often assume a static environment, where flight schedules and operational constraints are deterministic and predictable. While effective in stable conditions, such models fail to adequately account for the inherent uncertainties and disruptions prevalent in modern airport operations(Şeker & Noyan, 2012; Kim et al., 2023).

In summary, gate assignment is no longer just a tactical problem constrained to short-term objectives. It has evolved into a complex, multi-faceted challenge requiring integration with broader airport operations and the incorporation of uncertainty-aware strategies. By adopting advanced optimization methods and leveraging modern technologies, airports can enhance their ability to manage disruptions, improve resource utilization, and deliver a seamless passenger experience in an increasingly volatile and interconnected aviation environment.

1.2.3 The interdependence between airport slot allocation and gate assignment

The relationship between ASA and gate assignment is both complex and critical, as these two processes are inherently interdependent. Slot allocation determines the timing of aircraft arrivals and departures, which directly impacts gate availability. Conversely, gate constraints—such as size compatibility, turnaround requirements, and stand layout—can retroactively undermine the feasibility of allocated slots. For example, a wide-body aircraft requiring a specialized gate cannot be assigned a slot during peak hours if all compatible gates are occupied, regardless of its slot allocation. This interdependence highlights the need for a coordinated approach to avoid operational inefficiencies and disruptions.

Traditionally, ASA and gate assignment have been treated as separate problems, with slot allocation handled as a strategic, pre-season planning activity and gate assignment addressed as a short-term, operational decision-making process. This separation often leads to suboptimal outcomes, as slot allocation decisions made without considering gate constraints may result in schedules that are theoretically feasible but operationally fragile. For example, a perfectly allocated slot schedule may fail during execution if gate constraints at peak times are not considered, leading to delays, gate conflicts, or excessive aircraft taxiing.

Integrating ASA and gate assignment offers significant opportunities to enhance airport efficiency and operational resilience. By aligning long-term slot allocation with real-time gate assignment, airports can create more robust schedules that account for both strategic planning goals and dynamic operational constraints. For instance, a collaborative approach might involve reserving buffer gates during peak periods as contingency for high-priority flights, or dynamically adjusting gate assignments in response to real-time flight delays while preserving the integrity of pre-allocated slots.

Furthermore, disruptions in one process can create cascading effects in the other. A delayed flight occupying a gate longer than anticipated can impact subsequent gate assignments, potentially requiring slot adjustments to accommodate the disrupted schedule. Similarly, a misaligned slot allocation can lead to excessive gate conflicts, forcing last-minute operational adjustments that negatively affect passenger satisfaction and resource utilization. Addressing these ripple effects requires a holistic view of airport operations, where ASA and gate assignment are jointly optimized.

This thesis emphasizes the integration of ASA and gate assignment under uncertainty, proposing models that bridge the gap between strategic planning and real-time operations. By combining robust optimization for slot allocation with adaptive, uncertainty-aware gate assignment methods, the proposed framework aims to address the challenges of interdependence, ensuring both processes work harmoniously to improve airport efficiency and passenger experience.

1.3 Contributions

Despite extensive research on airport slot allocation (ASA), significant gaps remain in addressing operational uncertainties and integrating strategic planning with real-time decision-making. This thesis bridges these gaps by introducing innovative methodologies that holistically link long-term airport planning with short-term operational adaptability, validated through real-world case studies. The research proposes a novel modeling framework that integrates airport slot allocation with gate assignment under uncertainty. Unlike traditional approaches that address these problems in isolation, the framework connects strategic decisions about slot allocation with operational gate assignments, accounting for both probabilistic uncertainties, such as stochastic flight delays, and worst-case disruptions, such as extreme weather events. By combining strategic and operational aspects, this framework ensures a more adaptive and resilient airport management process.

We address these challenges with two models: a two-stage stochastic integer program (2SSIP) for scenario-based uncertainty and a two-stage robust optimization model for worst-case resilience. The stochastic model focuses on optimizing slot allocation in its first stage while dynamically adjusting gate assignments in the second stage through a rolling horizon approach, enabling real-time updates to mitigate disruptions caused by uncertainties such as flight delays. In contrast, the robust optimization model prioritizes ensuring operational feasibility under extreme disruption scenarios, acting as a safeguard against adverse conditions such as severe weather or cascading delays. These complementary paradigms jointly furnish a comprehensive approach to uncertainty, combining probabilistic adaptability with worst-case resilience to enhance airport operational efficiency and reliability.

To enhance computational efficiency, we develop specialized decomposition algorithms for both models. For the stochastic programming framework, a Benders-based decomposition effectively separates strategic planning from operational adjustments. We apply a nested column-and-constraint generation (C&CG) algorithm to the proposed robust model.

We validate the proposed methodologies through case studies based on historical flight data from Shanghai Hongqiao International Airport, one of China's busiest hubs. The findings show that the proposed frameworks are both practical and effective for addressing congestion, uncertainty, and resource constraints. By harmonizing long-term planning with real-time operational adjustments, the contributions of this thesis provide airports with a decision-making toolkit that enhances efficiency, robustness, and adaptability, offering actionable solutions for congested airports worldwide.

1.4 Outline

The thesis comprises two research articles addressing the integration of airport slot allocation (ASA) and gate assignment under operational uncertainty. Below is a summary of each chapter's focus:

Chapter 2 briefly review the literature on airport slot allocation and gate assignment.

Chapter 3 introduces a 2SSIP framework to jointly optimize long-term ASA decisions and real-time gate assignments, explicitly accounting for uncertainties such as flight delays. The first stage allocates slots to balance demand and capacity, while the second stage dynamically adjusts gate assignments using a rolling horizon mechanism as uncertainty unfolds. We employ Benders decomposition to decouple strategic slot allocation (master) from real-time gate assignment (subproblems), significantly reducing computational complexity. Validated with historical flight data from Shanghai Hongqiao International Airport, the model demonstrates improved operational efficiency, including reduced delays and enhanced gate utilization.

Chapter 4 complements this approach with a two-stage robust optimization framework that safeguards operations against worst-case disruptions, such as extreme weather or system failures. The first stage determines risk-averse slot allocations, while the second stage generates gate assignments resilient to maximum uncertainty scenarios. A C&CG algorithm solves large-scale instances efficiently, achieving 95% schedule feasibility under adversarial conditions for instances exceeding 500 daily flights. Tested against real-world data from the same airport, the framework minimizes worst-case delays and reconfiguration costs.

Chapter 5 concludes the major findings of the thesis and proposes topics for further study.

Chapter 2

Literature review

2.1 Airport slot allocation

Fan & Odoni (2002) reviewed demand management strategies implemented globally, particularly in the U.S. and Europe. These strategies fall into two main categories. Administrative strategies involve setting limits on airport capacity (runways, aprons, terminals, etc.) at various intervals, known as "declared capacities." Market mechanisms, including congestion pricing and slot auctions, are employed to curb airport congestion (see Wang et al. 2023b for details). Our study specifically focuses on the administrative strategy of slot allocation in airport capacity management.

The initial work on ASA by Zografos et al. (2012) provided a mathematical model aimed at minimizing total displacements. Zografos et al. (2018) expanded this approach by proposing two bi-objective models: one aimed at minimizing both total and maximum displacement, and another focused on minimizing the number of rejected slots and total displacement. Subsequent studies also incorporated fairness as an objective. Zografos & Jiang (2019) developed a bi-objective formulation that balances schedule efficiency and fairness. Jiang & Zografos (2021) introduced a voting mechanism that suggests the preferences of airport stakeholders. Fairbrother et al. (2020) introduced a displacement budget mechanism and a hierarchical slot allocation mechanism, integrating airline preferences and fairness considerations, while addressing slot requests exceeding available capacity.

Recent studies in airport slot allocation (ASA) have increasingly incorporated multiple objectives. Building on earlier treatments of priorities, Ribeiro et al. (2018) presented a new formulation grounded in priority rules. Building on this, Katsigiannis & Zografos (2023) proposed a valuation function to estimate the value of slot requests, integrating airlines' preferences into the decision-making process. Katsigiannis & Zografos (2021) introduced the Time Flexibility Indicator (TFI), which captures airlines' scheduling flexibility preferences and considers dynamic capacity at the apron and terminal. With a focus on passengers, Birolini et al. (2023) aimed to reduce both the incidence of lost passengers and interflight connection times. While prior research has examined efficiency, fairness, and stakeholder preferences in planning-level slot allocation, it often overlooks how flight schedules influence airport operations and congestion, particularly the role of operational-level decisions.

Another recent direction in ASA research is the extension from single airports to airport networks. Initially, Castelli et al. (2012) and Pellegrini et al. (2012) formulated network-level slot allocation models, focusing on minimizing displacement and maximizing accepted requests. Pellegrini et al. (2017) emphasized schedule regularity and quality by minimizing rejections and displacement. Considering the entire scheduling season is vital for maintaining schedule regularity (Zografos et al., 2012). Benlic (2018) introduced a "series of slots" constraint, while Fairbrother & Zografos (2021) divided the season into sub-periods to optimize slot requests. Unlike others, Keskin & Zografos (2023) proposed novel formulation that prioritizes schedule efficiency and fairness, using outcomes from individual airport allocations as inputs.

The ASA literature primarily consists of deterministic models, with limited research on modeling ASA under uncertainty. Some studies have explored the balance between efficient scheduling and network-wide delay, with capacity reductions identified as a key driver. Some researchers developed stochastic programming to account for airport capacity uncertainty caused by factors such as adverse weather conditions (Corolli et al., 2014; Wang & Jacquillat, 2020; Liu et al., 2022). Corolli et al. (2014) were the first to address the balance between scheduling efficiency and flight delays at network level, focusing on aggregate-level operating dynamics. Liu et al. (2022) proposed an ASA model at the network level, considering airport capacity uncertainty with different scenarios. Wang & Zhao (2020) introduced a robust optimization model for simultaneous slot allocation, and Wang et al. (2023a) studied a chance-constrained formulation for multiple-airport systems.

Most ASA research under uncertainty targets network-level problems, modeling airport capacity as a random variable in slot allocation. Operationally, these studies aim to minimize delay costs induced by planning-level flight schedules. There is limited research that has incorporated a single airport operational problem into an ASA model. In a two-stage stochastic programming framework at the network level, Wang & Jacquillat (2020) included the ground-holding problem as the second-stage decision. As far as the author is aware, existing ASA under uncertainty work does not consider specific operational issues in single-airport slot allocation.

2.2 Gate assignment

As a well-studied multiobjective optimization problem, gate assignment seeks to limit unassigned flights to apron stands, reduce tow movements, and minimize passenger walking distances. Comprehensive reviews are provided by Bouras et al. (2014) and Daş et al. (2020). We now provide a brief overview of the problem's stochastic variants.

Some stochastic perspective models flight arrival and departure times as random variables to handle gate assignment uncertainty. Yan & Tang (2007) formulated this problem as a multicommodity network flow, targeting the minimization of total passenger waiting time and adding penalties on connecting-flight arcs across delay scenarios. Şeker & Noyan (2012) modeled gate conflicts based on uncertainties in flight timings. Some researchers have employed robust optimization to manage stochastic delays by incorporating buffer times into gate departure schedules (Hassounah & Steuart, 1993; Yan & Chang, 1998). Additionally, Narciso & Piera (2015) evaluated the robustness of different strategies through simulations. Aoun & El Afia (2014) introduced an Markov Decision Process approach, using transition probabilities to capture potential gate conflicts arising from operational delays. Our proposed approach shares similarities by incorporating uncertainties in flight delays as model inputs, treating flight delays as random variables, and adopting a rolling horizon approach to better align with real airport operations.

Meanwhile, the broader gate assignment literature has continued to advance in both modeling and solution methods, which further underscores that gate assignment is a crowded research area. On the algorithmic side, metaheuristics and large neighborhood search have been widely used for large-scale instances (Cheng et al., 2012; Benlic et al., 2017). Recent studies also extend classical formulations by incorporating managerial considerations such as fairness (Jiang et al., 2023), and by explicitly improving robustness against conflicts under uncertainty through robust formulations and robust search heuristics (Du et al., 2023; Yu et al., 2017). In parallel, exact and hybrid approaches (e.g., matheuristics and branch-and-price) have been proposed to strengthen optimality and scalability for realistic problem sizes (Karsu et al., 2021; Karsu & Solyalı, 2023; Liu & Xiang, 2023; Okur et al., 2025), and learning-augmented heuristics have recently emerged to improve feasibility and solution quality in large instances (Li et al., 2022).

Despite these advances, most studies treat gate assignment as a standalone operational decision given an exogenous (fixed) flight timetable, and thus primarily improve gate-level performance without explicitly modeling how planning-stage schedule design reshapes downstream gate feasibility and costs. In contrast, this thesis positions gate assignment as a real-time rolling recourse decision under progressively revealed delay information and embeds it within a joint framework where planning-stage slot allocation decisions internalize expected operational consequences. This joint perspective distinguishes our work from existing stochastic/robust gate assignment studies: rather than optimizing gates for a given schedule, we quantify how slot-allocation decisions can proactively smooth demand peaks and reduce expected downstream gate-management costs under uncertainty, thereby converting planning-stage interventions into measurable operational benefits.

2.3 Application of stochastic modelling in air traffic management

Effective airport management and optimization require careful attention to both planning and operations. Because airports interact within a wider network and real-world operations are inherently uncertain, analyzing strategic decisions or operations in isolation yields limited insight. In recent decades, two-stage and multistage stochastic models have become increasingly common in Air Traffic Management (ATM), linking operational uncertainties to strategic decision-making. For an extensive review of stochastic modeling in ATM under uncertainty, see Shone et al. (2021). Below, we briefly summarize key studies.

ATM aims to manage air traffic safely and efficiently. A key operational tool within ATM is air traffic flow management.Andreatta & Romanin-Jacur (1987) were pioneers in using a stochastic optimization method to minimize total expected flight delay costs. Richetta & Odoni (1994) presented a stochastic programming model addressing the ground-holding problem at a single airport. Chandran & Balakrishnan (2007) were the first to incorporate uncertainty into the deterministic aircraft landing problem, adding probabilistic considerations. Solveling et al. (2011) developed a two-stage stochastic model for aircraft landing schedules, initially establishing a sequence of weight classes and then assigning flights to this sequence, considering weight compatibility and operational uncertainties. Based on the study, Solak et al. (2018) model second-stage costs using exact runway operation times instead of sequence-only information. Further real-world studies using two-stage stochastic programming have been conducted by Liu et al. (2018), Khassiba et al. (2020), and Khassiba et al. (2022). While the application of stochastic programming in ATM has been increasing, its utilization in ASA and gate assignment remains limited.

For long-term planning in ATM, one key aspect is ASA, which is discussed in Section 2.1. To address the stochastic nature of ASA problems, several studies have proposed two-stage stochastic programming as a suitable method. Corolli et al. (2014) focused on airport slot allocation within a network, aiming to minimize total displacement from slot allocation and anticipated operational delay costs. They were the first to incorporate time slot allocation and the stochastic capacity of airports, using the Sample Average Approximation method to manage uncertainty. However, their approach focused solely on an aggregate model at the operational level, neglecting the operational dynamics of the flight hierarchy. Wang & Jacquillat (2020) expanded the scope by examining the entire U.S. network and presenting a two-stage stochastic programming framework with integer recourse. They introduced dual integer cuts to accelerate the decomposition algorithm, leveraging the integer nature of second-stage decision variables. Liu et al. (2022) proposed a network-level ASA model that accounts for capacity uncertainty and solved it using Benders decomposition.

Recent work increasingly acknowledges the interdependence between long-term planning and short-term operations. These models treat airport capacity as uncertain in stochastic operational scenarios and primarily seek to minimize network-wide delay costs, reducing delays and their propagation across airports. However, focusing solely on delay avoidance does little to improve ground-side efficiency, such as apron layout planning, gate assignment, and related tasks.

The thesis targets a single airport, jointly optimizing slot allocation and gate assignment. Our approach includes a 2SSIP formulation and a two-stage robust optimization formulation.

Chapter 3

Joint Optimization of Airport Slot Allocation and Gate Assignment: A Stochastic Integer Programming Approach

3.1 Introduction

The Statistical Bulletin on the Development of the China Civil Aviation Industry in 2023 reports a strong rebound in passenger and cargo traffic, driven by the gradual easing of COVID-19 impacts and rapid industry growth (CAAC, 2024). This recovery signals continued expansion. However, rising demand can exacerbate congestion—manifested as lower punctuality and more delays. While disruptions are often triggered by weather or airline operations, a fundamental driver is the widening gap between the rising air travel demand and limited airport capacity (Zeng et al., 2021).

Given the significant time and cost of expanding capacity, it is essential to optimize planning and operations using existing airport resources. Airport slot allocation (ASA) is a key lever for improving operational efficiency. ASA is a (relatively) long-term planning decision that should address demand-capacity imbalances in congested level 3 airports (IATA, 2024). It involves collaborative efforts among various stakeholders, including slot coordinators, airports, and airlines, to develop and adjust flight schedules while ensuring compliance with airport coordination parameters. The ASA process is carried out twice a year to determine the flight schedule for the following operating season. The decision-making process typically begins about six months before the operating period. During the initial slot allocation stage, coordinators receive slot requests from airlines for airport resources. Based on the World Airport Slot Guidelines (WASG), coordinators allocate available slots to these requests. Determining flight schedules for the subsequent bi-annual slot coordination conferences forms the core of the ASA process. Once finalized at these conferences, the flight schedule is implemented for actual operations in the extended scheduling season, usually lasting six months, and remains largely unchanged during this period.

Meticulous scheduling notwithstanding, actual operations commonly diverge from the intended timetable owing to unexpected disruptions like mechanical failures or severe weather. Overloading the schedule with excessive flights within a limited timeframe can worsen airport congestion if these flights experience delays due to operational uncertainties. Given the networked nature of air traffic, significant airport delays mean that postponing one early flight can induce widespread downstream disruptions. In 2023, roughly 13.2% of flights in China experienced delays averaging 10 minutes. Weather accounted for 60.42% of delays, and airline-related factors for 14.68% (IATA, 2024). When substantial disruptions occur at an airport, it becomes necessary to revise flight schedules to align with real-time operational capabilities. This revision involves delaying and canceling flights, which may further cause a decrease in efficiency and loss of profit. Therefore, besides considering capacity constraints and management efficiency, including the consideration of uncertainties occurring at the operational level during the ASA process have the potential to further enhance overall performance.

Gate assignment is a critical short-term airport management task, and it is frequently subject to uncertainty. Gates play a vital role in facilitating various ground services for arriving flights, including boarding, maintenance, and deplaning (Evler et al., 2021) making them a limited airport resource that requires effective utilization. Typically, airports assign gates to arriving flights based on established flight schedules. However, uncertainties occurring in flight delays often disrupt the execution of the gate assignment plan, leading to frequent reassignments during operations. Such actions have a negative impact on flight punctuality and passenger satisfaction.

Current approaches to optimizing airport scheduling and operations often ignore the interdependence and mutual influence between planning-level ASA and operational-level gate assignment. However, it is important to recognize that ground activities and flight schedules are closely intertwined, as gate assignment is determined by flight schedules, which are influenced by ASA. Similarly, considering uncertainties in gate assignments while making decisions during ASA has the potential to enhance operational efficiency, reduce delays, and optimize resource utilization at airports. Therefore, in this context, we address the optimization of ASA at a planning-level in gate assignment at operational level while considering uncertainties.

The main contributions are as follows: (i) We introduce a modeling framework that optimizes long-term slot allocation while explicitly accounting for uncertainty in short-term gate assignment at a single airport. (ii) We formulate the joint problem as a 2SSIP: the first-stage optimizes slot allocation, and the second-stage models gate assignment as a real-time resource allocation problem under uncertainty and capacity constraints. (iii) We design a tailored Benders decomposition with optimality cuts and show it is computationally efficient. (iv) Using historical data from Shanghai Hongqiao International Airport, we conduct realistic numerical experiments that demonstrate the method's effectiveness and scalability, underscoring its potential for airport scheduling and operations.

The remainder of the chapter is organized as follows. Section 3.2 presents the notation, problem description, and the 2SSIP model for the joint problem. Section 3.3 demonstrates the proposed algorithm. Section 3.4 reports numerical studies evaluating the algorithm and the model's effectiveness. Section 3.5 concludes by synthesizing the main findings and highlighting future research.

3.2 Problem description

We present the problem statement and notation in Section 3.2.1, and provide the model formulation in Section 3.2.2.

3.2.1 Notation and problem statement

The section addresses the joint optimization of two interdependent airport problems: planning-level slot allocation and operational-level gate assignment. The model takes as inputs: (i) airlines' slot requests, (ii) airport capacity, and (iii) operating scenarios. Preferred schedules are derived from airlines' slot requests. Capacity includes coordination parameters and the number and types of gates. Coordination parameters specify per-period limits on arrivals and departures, reflecting runway, apron, and terminal capacities. Scenarios capture realized flight delays under varying operating conditions.

In practice, airport "declared capacities" are often enforced over multiple rolling time windows (e.g., per 15 minutes, per 60 minutes), rather than only at the base discretization granularity. We therefore define C as the set of airport capacity timescales (window lengths) expressed in the same unit as the discretized time grid T. For each start interval $t \in T$ and timescale $c \in C$, the runway capacity parameter $u_{t,c}^{\text{Arr/Dep/Total}}$ specifies the maximum number of arrivals, departures, or total movements allowed within the window $[t, t+c)$. This sliding-window representation captures operational rules such as "at most n_c departures in any c-minute window."

We introduce a two-stage procedure in which (i) ASA allocates arrival and departure times to produce a feasible flight schedule, and (ii) a gate assignment model assigns gates and measures the resulting performance.

In the first stage, ASA minimizes (i) rejected slot requests and (ii) total deviation from airlines' preferred schedules, subject to capacity constraints. While slot requests may specify only one leg, we assume each request bundles both arrival and departure for the same aircraft, consistent with standard practice.

Gate assignment is made after operational uncertainty is realized. In the second stage, given realized arrival/departure delays, we assign gates to minimize total parking cost while enforcing non-overlapping gate usage within each scenario. Unlike conventional static approaches, we model gate assignment as a dynamic, continuously updated process under unfolding uncertainties. See Figure 3.1 for an illustration.

At time interval t, due to uncertainty, aircraft arrival and departure times are known with relative accuracy only τ time intervals in advance. This follows real-world observations, which typically show minimal variation in flight times between fixed origin-destination pairs. Once an aircraft departs from its previous airport, its arrival time at the destination becomes known, allowing its departure to be scheduled according to the current airport situation. Figure 3.1 illustrates this with aircraft F_1, F_2, ..., F_9, where the horizontal axis represents time. Gates G_1, G_2, ..., G_n are shown, and "At time interval x" denotes the gate assignment plan for each time interval t_x. The figure presents the gate assignment decision at three different time intervals, each within a time window $[t, t+\tau]$, marked by a red border.

Aircraft are categorized based on their status: those that have left the airport, those that have arrived but not yet departed, those en route with known exact arrival times, and those en route with unknown exact arrival times. Whether an aircraft's exact arrival time is known depends on the interval $[t, t+\tau]$. As per our assumption, the exact arrival time is known τ intervals in advance. In Figure 3.1, the statuses of F_2, F_3, F_4, F_5, and F_6 change over time at intervals t_1 and t_2. Due to uncertainty, the arrival times of F_5, F_6 and F_7 in t_1's plan are unknown, with G_1 assigned to F_5 and G_2 to F_6 and F_7. By t_2, the gate assignment plan optimizes, assigning G_2 to F_5 and G_1 to F_6 and F_7.

Thus, the dynamic process of gate assignment follows these rules: at interval t, the uncertainty about actual arrival times before $t+\tau$ is resolved, and gate assignments for these aircraft remain unchanged in subsequent plans. For other aircraft, gate assignments are based on flight schedule and may adjust in future plans.

In this problem, we focus on slot allocation and gate assignment for a representative operating day within the scheduling season. The planning horizon is discretized into a set of time intervals T. Consistent with the case-study setting, we divide one day into 288 intervals of 5 minutes each, i.e., $T = \{0, \ldots, 287\}$, which provides a unified temporal grid for

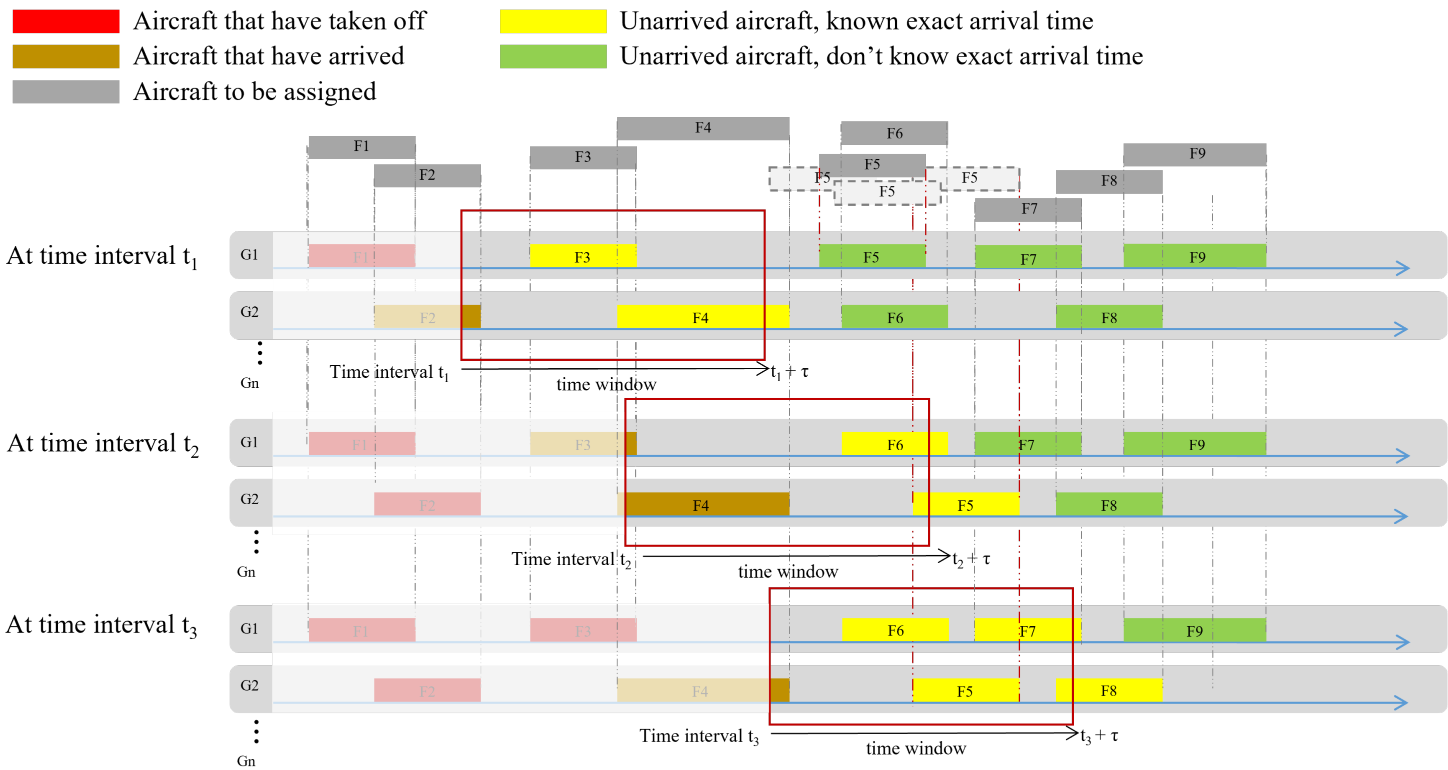


Figure 3.1: The dynamic process of airport gate assignment

both the planning-stage slot allocation decisions and the operational-stage gate assignment evaluation.

In summary, the joint ASA and gate assignment problem proceeds as follows. In the first-stage, ASA allocates each aircraft's arrival and departure at a single airport. In the second-stage, given those slots, gates are assigned while accounting for real-time operational uncertainties. To capture the interdependence between stages, we adopt a 2SSIP approach that jointly optimizes slot allocation and gate assignment. The model aims to minimize rejected slot requests, total schedule displacement, and the expected second-stage cost across scenarios reflecting arrival/departure uncertainty.

3.2.2 Model formulation

We build on the notation and ASA formulation of Katsigiannis & Zografos (2021) to model the joint ASA–gate assignment problem as a 2SSIP. Table 3.1 summarizes the notation.

Table 3.1: Notation for model formulation

Notation	**Description**
Sets	
General sets	
F	Set of aircraft, $i \in F$
T	Set of time intervals, $t \in \{0, ..., \|T\|\}$
First-stage	
L	Set of airport terminals, $l \in L$
C	Set of airport capacity timescales, $c \in C$
S	Set of service types, $s \in \{Pax, Cgo\}$, which is divided into passenger (Pax) and cargo (Cgo) services.
Second-stage	
Ω	Set of scenarios, $\omega \in \Omega$
G	Set of gates, $g \in G$

Notation	Description
Parameters	
First-stage	
ζ	Penalty cost for rejecting a slot request
$r_{i,a(d)}$	Requested time slots of aircraft i, categorized as $r_{i,a}$ (arrival) and $r_{i,d}$ (departure)
$T_{\min,i}$, $T_{\max,i}$	Minimum and maximum turnaround time of aircraft i
$\beta_{i,s}^{\text{Arr(Dep)}}$	Equals 1 if aircraft i's arrival (departure) is of service type s, 0 otherwise
$u_{t,c}^{\text{Arr(Dep,Total)}}$	Runway capacity for arrivals (departures, or total) within the time window $[t,t+c)$ for a time interval of length c
$\delta_{t,s}^{l}$	Apron capacity for aircraft of service type s at time interval t on terminal l
$E_{t,c}^{\text{Arr(Dep)}}$	Passenger terminal capacity for arrivals (departures) during time window $[t,t+c)$ for a time interval of length c
e_i	Number of seats requested for aircraft i
$\psi_{i,t}^{\text{Arr(Dep)}}$	Displacement of time slot request for aircraft i: $\psi_{i,t}^{\text{Arr}} = \lvert t - r_{i,a}\rvert$, $\psi_{i,t}^{\text{Dep}} = \lvert t - r_{i,d}\rvert$
Ψ	Maximum displacement of all aircraft
λ_i	Equals 1 if the arrival and departure of aircraft i must be assigned to the same terminal; 0 otherwise
$\pi_i^{\text{Arr(Dep)}}$	Load factor for arrival (departure) of aircraft i
$\varepsilon_{i,l}^{\text{Arr(Dep)}}$	Equals 1 if aircraft i can arrive (depart) at terminal l, 0 otherwise
Second-stage	
σ	Penalty cost of parking at temporary apron (Gate 0)
$\hat{a}_i^t(\omega)$	Random parameter: deviation of the real arrival time a_i^t of aircraft i from the scheduled time at interval t under scenario $\omega \in \Omega$
$\hat{d}_i^t(\omega)$	Random parameter: deviation of the real departure time d_i^t of aircraft i from the scheduled time at interval t under scenario $\omega \in \Omega$

Notation	Description
η	Minimum buffer time between two successive aircraft parked at the same gate
$o_{i,g}$	Equals 1 if aircraft i can park at gate g, 0 otherwise
$z_{i,g}^{b,t}(\omega)$	Equals 1 if gate g is assigned to aircraft i at the start of time interval t (unchangeable) in scenario $\omega \in \Omega$, 0 otherwise
Variables and expressions	
First-stage	
$x_{i,t}^{l}$	Equals 1 if aircraft i is allocated to arrive at terminal l at time t, 0 otherwise
$y_{i,t}^{l}$	Equals 1 if aircraft i is allocated to depart from terminal l at time t, 0 otherwise
μ_i	Equals 1 if aircraft i is allocated both an arrival and departure time slot, 0 otherwise
$n_{t,s}^{l}$	Number of service type s aircraft parked at terminal l at time interval t
Second-stage	
$z_{i,g}^{t}(\omega)$	Equals 1 if gate g is assigned to aircraft i at time interval t in scenario $\omega \in \Omega$, 0 otherwise
$z_{i,g}^{e,t}(\omega)$	Equals 1 if gate g is assigned to aircraft i at the end of time interval t in scenario $\omega \in \Omega$, 0 otherwise
$a_i^t(\omega)$	Arrival time of aircraft i at time interval t in scenario $\omega \in \Omega$
$d_i^t(\omega)$	Departure time of aircraft i at time interval t in scenario $\omega \in \Omega$
$p_{i,t}^{b}(\omega)$	The start time of the aircraft's parking at the gate at time interval t in scenario $\omega \in \Omega$
$p_{i,t}^{e}(\omega)$	The end time of the aircraft's parking at the gate at time interval t in scenario $\omega \in \Omega$

The first-stage ASA model is defined in Eqs. (3.1)–(3.20). In this stage, the objective minimizes (i) the number of rejected slot requests, (ii) total displacement, and (iii) the expected second-stage gate assignment cost across all scenarios. The second-stage treats gate assignment as a real-time problem (Eqs. (3.22)–(3.34)), minimizing total ground delay and the cost of assigning aircraft to the temporary apron. Gate 0 represents a temporary apron with assumed infinite capacity. An offline variant of the real-time gate assignment problem (Eqs. (3.35)–(3.45)) is also developed. This offline model is used to generate the dual offline cuts in (3.49) and to assess the performance of the rolling-horizon approach.

$$\min \zeta \left(|F| - \sum_{i \in F} \mu_i \right) + \rho \sum_{l \in L} \sum_{i \in F} \left[\sum_{t \in T} x^l_{i,t} \psi^{Arr}_{i,t} + \sum_{t \in T} y^l_{i,t} \psi^{Dep}_{i,t} \right] + (1-\rho) E\left[h(\nu, \xi)\right] \quad (3.1)$$

subject to General constraints:

$$\sum_{l \in L} \sum_{t \in T} x^l_{i,t} \leq 1 \quad \forall i \in F \quad (3.2)$$

$$\sum_{l \in L} \sum_{t \in T} y^l_{i,t} \leq 1 \quad \forall i \in F \quad (3.3)$$

$$\sum_{l \in L} \sum_{t \in T} x^l_{i,t} = \sum_{l \in L} \sum_{t \in T} y^l_{i,t} = \mu_i \quad \forall i \in F \quad (3.4)$$

$$\left(\sum_{t \in T} x^l_{i,t} - \sum_{t \in T} y^l_{i,t} \right) (1 - \lambda_i) = 0 \quad \forall l \in L, i \in F \quad (3.5)$$

$$\mu_i T_{\min,i} \leq \sum_{l \in L} \sum_{t \in T} y^l_{i,t} t - \sum_{l \in L} \sum_{t \in T} x^l_{i,t} t \leq \mu_i T_{\max,i} \quad \forall i \in F \quad (3.6)$$

$$\sum_{l \in L} \sum_{t \in T} x^l_{i,t} \psi^{Arr}_{i,t} \leq \Psi \quad \forall i \in F \quad (3.7)$$

$$\sum_{l \in L} \sum_{t \in T} y^l_{i,t} \psi^{Dep}_{i,t} \leq \Psi \quad \forall i \in F \quad (3.8)$$

Runway capacity constraints:

$$\sum_{i \in F} \sum_{t' \in [t, t+c-1]} \sum_{l \in L} x^l_{i,t'} \leq u^{Arr}_{t,c} \quad \forall c \in C, t \in [0, |T| - c] \quad (3.9)$$

$$\sum_{i \in F} \sum_{t' \in [t, t+c-1]} \sum_{l \in L} y^l_{i,t'} \leq u^{Dep}_{t,c} \quad \forall c \in C, t \in [0, |T| - c] \quad (3.10)$$

$$\sum_{i \in F} \sum_{t' \in [t, t+c-1]} \sum_{l \in L} \left(x^l_{i,t'} + y^l_{i,t'} \right) \leq u^{Total}_{t,c} \quad \forall c \in C, t \in [0, |T| - c] \quad (3.11)$$

Apron capacity constraints:

$$\sum_{i\in F} x^l_{i,t}\beta^{Arr}_{i,s} - \sum_{i\in F} y^l_{i,t}\beta^{Dep}_{i,s} + n^l_{t-1,s} = n^l_{t,s} \quad \forall l\in L, s\in S, t\in T/\{0\} \tag{3.12}$$

$$\sum_{i\in F} x^l_{i,0}\beta^{Arr}_{i,s} - \sum_{i\in F} y^l_{i,0}\beta^{Dep}_{i,s} = n^l_{0,s} \quad \forall l\in L, s\in S \tag{3.13}$$

$$n^l_{t,s} \le \delta^l_{t,s} \quad \forall l\in L, t\in T, s\in S \tag{3.14}$$

Passenger terminal capacity constraints:

$$\sum_{l\in L}\sum_{i\in F}\sum_{t'\in[t,t+c-1]} x^l_{i,t'} e_i \pi^{Arr}_i \le E^{Arr}_{t,c} \quad \forall t\in[0,|T|-c] \tag{3.15}$$

$$\sum_{l\in L}\sum_{i\in F}\sum_{t'\in[t,t+c-1]} y^l_{i,t'} e_i \pi^{Dep}_i \le E^{Dep}_{t,c} \quad \forall t\in[0,|T|-c] \tag{3.16}$$

Terminal compatibility constraints:

$$\sum_{t\in T} x^l_{i,t} \le \varepsilon^{Arr}_{i,l} \quad \forall l\in L, i\in F \tag{3.17}$$

$$\sum_{t\in T} y^l_{i,t} \le \varepsilon^{Dep}_{i,l} \quad \forall l\in L, i\in F \tag{3.18}$$

Decision variables:

$$\mu_i \in \{0,1\} \quad \forall i\in F \tag{3.19}$$

$$x^l_{i,t}, y^l_{i,t} \in \{0,1\} \quad \forall l\in L, i\in F, t\in T \tag{3.20}$$

$$n^l_{t,s} \in Z^+ \cup \{0\} \quad \forall l\in L, t\in T, s\in S \tag{3.21}$$

where $E[h(\nu,\xi)]$ denotes the expected recourse, with $h(\nu,\xi)$ being the optimal second-stage objective. The the first-stage decision vector is $\nu = \left\{\mu_i, x^l_{i,t}, y^l_{i,t},\right\}$.

In the first stage, the objective function (3.1) minimizes (i) the number of rejected slot requests, (ii) total displacement weighted by ρ, and (iii) the expected second-stage gate assignment cost weighted by $1-\rho$.

There are five groups of constraints. The first, given by constraints (3.2)-(3.6) address the general feasibility and integrity of the ASA problem. Specifically, constraints (3.2)-(3.3) ensure that each aircraft's departure (arrival) slot request is either allocated to at most one slot and one terminal, or rejected. Constraints (3.4) ensure that the acceptance or rejection of an aircraft's arrival slot matches its departure slot status. To ensure terminal consistency, constraints (3.5) ensure that aircraft arrivals and departures occur at the same terminal when necessary. Constraints (3.6) enforce minimum and maximum turnaround times unless the aircraft is rejected. Constraints (3.7) and (3.8) ensure maximum displacement of each aircraft arrival and departure. Runway capacity constraints (3.9)-(3.11) restrict aircraft movements to not exceed runway capacity at any time interval. For apron capacity, the number of parked aircraft ($n^l_{t,s}$) is calculated using the parameter $\beta^{Arr(Dep)}_{i,s}$. Constraints (3.12)-(3.13) define the parked-aircraft count at each interval as the sum of those already parked plus arrivals and departures during that interval. Constraint (3.14) ensure that the apron capacity for each service type is not exceeded. Passenger terminal capacity constraints (3.15)-(3.16) use the weighting factor $\pi^{Arr(Dep)}_{i}$ to estimate passenger numbers, ensuring these do not exceed terminal capacity. Terminal compatibility is handled by constraints (3.17)-(3.18), which specify whether aircraft are allowed to park at particular terminals.

Stochastic airport conditions are modeled using a finite set of scenarios Ω. Let $\xi(\omega) = \left\{\hat{a}^t_i(\omega), \hat{d}^t_i(\omega)\right\}$ represent flight delays under scenario $\omega \in \Omega$. The second-stage online gate assignment model is expressed by $h(\nu, \xi(\omega))$ and is formulated by Eqs. (3.22)-(3.32).

Online Model$($*Online Gate Assignment at Time Interval t*$)$:

$$h(\nu, \xi(\omega)) = \min \sum_{i \in F} \left(p_{i,t}^{b}(\omega) - a_i^t(\omega) \right) + \left(p_{i,t}^{e}(\omega) - d_i^t(\omega) \right) + \sigma z_{i,0}^{t}(\omega) \tag{3.22}$$

subject to:

$$\sum_{g \in G} z_{i,g}^{t}(\omega) = \mu_i \quad \forall i \in F \tag{3.23}$$

$$\sum_{l \in L} \sum_{t' \in T} x_{i,t}^{l} t' + \hat{a}_i^t(\omega) = a_i^t(\omega) \quad \forall i \in F \tag{3.24}$$

$$\sum_{l \in L} \sum_{t' \in T} y_{i,t}^{l} t' + \hat{d}_i^t(\omega) = d_i^t(\omega) \quad \forall i \in F \tag{3.25}$$

$$p_{i,t}^{b}(\omega) \geq a_i^t(\omega) \quad \forall i \in F \tag{3.26}$$

$$p_{i,t}^{e}(\omega) \geq d_i^t(\omega) \quad \forall i \in F \tag{3.27}$$

$$p_{i,t}^{e}(\omega) \geq p_{i,t}^{b}(\omega) + T_{\min,i} \quad \forall i \in F \tag{3.28}$$

$$p_{i,t}^{e}(\omega) + \eta - p_{j,t}^{b}(\omega) \leq M\left(2 - z_{i,g}^{t}(\omega) - z_{j,g}^{t}(\omega)\right) \quad \forall i, j \in F,\, g \in G \backslash \{0\},\, i < j \tag{3.29}$$

$$z_{i,g}^{t}(\omega) \leq o_{i,g} \quad \forall i \in F, g \in G \tag{3.30}$$

$$z_{i,g}^{b,t}(\omega) \leq z_{i,g}^{t}(\omega) \quad \forall i \in F, g \in G \tag{3.31}$$

$$z_{i,g}^{t}(\omega) = z_{i,g}^{e,t}(\omega) \quad \forall i \in F, g \in G \tag{3.32}$$

$$z_{i,g}^{t}(\omega), z_{i,g}^{e,t}(\omega) \in \{0,1\} \quad \forall i \in F, g \in G \tag{3.33}$$

$$a_i^t(\omega), d_i^t(\omega), p_{i,t}^{b}(\omega), p_{i,t}^{e}(\omega) \in R^{+} \quad \forall i \in F \tag{3.34}$$

With ν fixed and $\xi(\omega)$ realized, the second-stage objective (3.22) minimizes the ground-delay cost and the cost of using the temporary apron. Ground delay has two components: (i) the difference between the gate parking start and the aircraft's actual arrival, and (ii) the difference between the gate parking end time and the aircraft's actual departure.

Once aircraft i is accepted in the first stage ($\mu_i = 1$), constraint (3.23)) ensures it is assigned to at most one gate. Scenario-dependent arrival and departure times for aircraft i in interval t are defined in constraints (3.24)–(3.25). Constraint (3.26) ensures the parking start time $p_{i,t}^{b}$ is no earlier than the arrival time a_i^t, and constraint (3.27) requires the parking end time $p^e_{i,t}$ be no earlier than the departure time d^t_i. Constraint (3.28) enforces minimum turnaround time. Constraint (3.29) prevents overlaps at regular gates (excluding Gate 0) by imposing a buffer η between consecutive aircraft. Aircraft–gate compatibility is enforced by constraint (3.30). Constraint (3.31) fixes pre-existing gate assignments at the start of interval t. Finally, constraint (3.32) sets the gate-assignment plan $z^{e,t}_{i,g}$ determined at the end of interval t.

Accordingly, we propose an offline version of the gate assignment model. In the offline version, we consider that for each $\omega \in \Omega$, the uncertainty information in the whole time period $\mathbf{T}$ has been revealed as input. The model is defined as follows:

Offline Model (*Offline Gate Assignment*):

$$h(\nu,\xi(\omega)) = \min \sum_{i\in F} \left(p^b_i(\omega) - a_i(\omega)\right) + \left(p^e_i(\omega) - d_i(\omega)\right) + \sigma z_{i,0}(\omega) \tag{3.35}$$

subject to:

$$\sum_{g\in G} z_{i,g}(\omega) = \mu_i \quad \forall i \in F \tag{3.36}$$

$$\sum_{l\in L}\sum_{t'\in T} x^l_{i,t'}t' + \hat{a}_i(\omega) = a_i(\omega) \quad \forall i \in F \tag{3.37}$$

$$\sum_{l\in L}\sum_{t'\in T} y^l_{i,t'}t' + \hat{d}_i(\omega) = d_i(\omega) \quad \forall i \in F \tag{3.38}$$

$$p^b_i(\omega) \geq a_i(\omega) \quad \forall i \in F \tag{3.39}$$

$$p^e_i(\omega) \geq d_i(\omega) \quad \forall i \in F \tag{3.40}$$

$$p^e_i(\omega) \geq p^b_i(\omega) + T_{\min,i} \quad \forall i \in F \tag{3.41}$$

$$p^e_i(\omega) + \eta - p^b_j(\omega) \leq M\left(2 - z_{i,g}(\omega) - z_{j,g}(\omega)\right) \quad \forall i,j \in F, g \in G\backslash\{0\}, i<j \tag{3.42}$$

$$z_{i,g}(\omega) \leq o_{i,g} \quad \forall i \in F, g \in G \tag{3.43}$$

$$z_{i,g}(\omega) \in \{0,1\} \quad \forall i \in F, g \in G \tag{3.44}$$

$$a_i(\omega), d_i(\omega), p^b_i(\omega), p^e_i(\omega) \in R^+ \quad \forall i \in F \tag{3.45}$$

Compared to the online model, the offline model removes the superscript t from the variables, indicating that the delay information for the aircraft does not reveal itself over time and the gate assignment plan does not change. When solving the online model using the rolling horizon framework, the online model reduces to the offline model when $\Delta t = T$, $\tau = T$, detailed in Section 3.3.3. Thus constraints (3.36)-(3.43) in the offline model are correspond to constraints (3.23)-(3.30) in the online model.

The proposed 2SSIP model size depends on the number of slot requests, terminals, time steps, capacity-interval lengths, and gates. In the numerical study discussed in Section 3.4.4, the model encompasses over 20 million variables and more than 6 billion constraints. These large numbers present substantial challenges for problem-solving. Consequently, Section 3.3 introduces a Benders-based decomposition algorithm specifically designed to efficiently address real-world, large-scale problems.

3.3 Solution algorithm

This section presents a Benders-based decomposition algorithm to address the computational challenges of the proposed 2SSIP (see Section 3.2.2). The algorithm optimizes slot allocation and gate assignment by decomposing the 2SSIP into the master problem (MP) and the subproblem (SP). The MP is formulated as:

$$\min \zeta \left(|F| - \sum_{i \in F} \mu_i \right) + \rho \sum_{i \in F} \left[\sum_{t \in T} x_{i,t} \psi_{t,a} + \sum_{t \in T} y_{i,t} \psi_{t,d} \right] + (1-\rho)\,\theta \tag{3.46a}$$

$$\text{s.t. } A\mu + Cx + Dy \leq b \tag{3.46b}$$

$$Q(v) \leq \theta \tag{3.46c}$$

$$x \in \{0,1\}^{n_1} \tag{3.46d}$$

$$y \in \{0,1\}^{n_2} \tag{3.46e}$$

$$\mu \in \{0,1\}^{n_3} \tag{3.46f}$$

where $Q(\nu)$ denotes the function of the vector parameter $\nu = \left\{\mu_i, x_{i,t}^l, y_{i,t}^l\right\}$. Constraint (3.46c) and the objective function term θ ensure that any optimal solution of the MP satisfies $Q(\nu) = \theta$.

Given that the expectation function $E\left[h\left(\nu, \xi\left(\omega\right)\right)\right]$ is not closed-form, it is approximated using the Sample Average Approximation (SAA) method. Thus, we have $Q(\nu) = E\left[h\left(\nu, \xi\left(\omega\right)\right)\right] = \frac{1}{N}\sum_{n=1}^{N} h\left(\nu, \xi_n\left(\omega\right)\right)$

For each $\omega \in \Omega$, the SP is given in standard form as:

$$h(\nu, \xi(\omega)) = \left\{ q^T z(\omega) + c^T p(\omega) \;\middle|\; \begin{array}{l} Wz(\omega) + Dp(\omega) = h(\omega) - T\nu, \\ z(\omega) \in \{0,1\},\ p(\omega) \geq 0 \end{array} \right\} \tag{3.47}$$

For this Section, Section 3.3.1 outlines the proposed two-stage stochastic programming solution algorithm. Section 3.3.2 introduces the optimal cuts for the proposed Benders-based algorithm. Section 3.3.3 presents the rolling horizon approach for solving the subproblem, including: (i) an online gate assignment by using the rolling horizon framework over the full time horizon, and (ii) a tabu search heuristic for single gate assignment subproblem. Section 3.3.4 discusses accelerating convergence of the Benders-based algorithm using neighborhood constraints. Finally, Section 3.3.5 describes the full algorithm.

3.3.1 Overview of solution algorithm

The approach employs Benders-based decomposition, with optimality cuts, as detailed in Section 3.3.2, as the primary mechanism to approximate the recourse function (3.22). Given that the SP is an integer programming problem, solving it via the rolling horizon approach is computationally intensive. Therefore, the approximate recourse function requires a large number of iterations. To address this, we devise the dual offline cut (3.49) based on the feasible solution of the MP and the relaxed form of the offline SP, thereby tightening the lower bound on θ and integrating it with the standard integer optimality cut (3.48).

The algorithm framework is based on the integer L-shaped method proposed by Laporte

& Louveaux (1993). Initially, the lower bound **L** for $Q(v)$ is calculated, which is zero when no aircraft are accepted in the first-stage decisions ($\sum_{i\in F}\mu_i = 0$). Subsequently, the solver resolves the MP with optimality cuts from previous iterations to derive the first-stage solution v and the corresponding 2SSIP's lower bound. The rolling horizon framework, Algorithm 1, is then applied to solve the SP in each scenario $\omega \in \Omega$, determining whether to add optimality cuts and neighborhood constraints, and updating the 2SSIP's upper bound. Notably, for any feasible solution of the MP, it is applicable to all SPs, as ensured by Constraint (3.26) for the online model and Constraint (3.41) for the offline model, which stipulate that if aircraft i is not accepted in the first stage of the master problem ($\mu_i = 0$), it will not be assigned to a gate. The comprehensive algorithm is presented in Section 3.3.5.

3.3.2 Optimality cuts

Given v^* is the current feasible solution of MP, let $S(v^*) = \{i : v_i^* = 1\}$. According to Laporte & Louveaux (1993), standard optimality cut at v^* of the online model is as follow:

$$\theta \geq (Q(v^*) - L)\left(\sum_{i\in S(v^*)} v_i - \sum_{i\notin S(v^*)} v_i - |S(v^*)|\right) + Q(v^*) \tag{3.48}$$

The enumerative nature of the optimality cut (3.48) guarantees that θ will eventually approximate $Q(v)$. However, relying solely on the optimality cut (3.48) is costly. Once the current feasible solution v^* is obtained, the rolling horizon approach is used to compute $Q(v^*)$, which entails solving a sequence of $\lceil T/\Delta t\rceil$ subproblems, where T is the whole time horizon and Δt is the step size. Details of the rolling horizon approach appear in Section 3.3.3. Since each subproblem is NP-hard, solving all subproblems to optimality can be time-consuming. This encourages using an offline version of the subproblem to generate cuts that strengthen the global lower bound on $Q(v)$.

When $Q(v)$ equals the integer-programming value given v, cuts can be constructed by solving the dual of the subproblem's LP relaxation. Because the online model's optimal value is typically no smaller than the offline model's due to stricter constraints (Glomb et al., 2022), we consider the LP relaxation of the offline SP.

Let $Q_{LP}^{\omega}(v^*)$ be the optimal value of dual offline subproblem in scenario $\omega \in \Omega$ at the current solution v^*. The first-stage decision variables appear in constraints (3.41)–(3.43) and thus require the corresponding dual multipliers. Let the dual variables associated with constraints (3.41)–(3.43) at v^* by $\alpha_{i,\omega}^*$, $\beta_{i,\omega}^*$, and $\gamma_{i,\omega}^*$, respectively. The Dual Offline Cut (DOC) is given by:

$$\theta \geq \sum_{\omega=0}^{N} Q_{LP}^{\omega}(v^*) + \alpha_{i,\omega}^*(\mu_i - \mu_i^*) + \beta_{i,\omega}^* \left(\sum_{t' \in T} t' \left(x_{i,t} - x_{i,t}^* \right) \right) + \gamma_{i,\omega}^* \left(\sum_{t' \in T} t' \left(y_{i,t} - y_{i,t}^* \right) \right) \tag{3.49}$$

where N is the number of scenarios. Equation (3.49) provides a tight lower bound on $Q(v)$, which is more advantageous in obtaining a feasible solution more quickly during the MP solution process.

3.3.3 Rolling horizon approach for solving subproblem

3.3.3.1 Rolling horizon framework

The proposed airport online gate assignment plan is implemented using the Rolling Horizon Approach (RHA) framework. In this framework, Δt denotes the optimization time step, while T denotes the time horizon length in the current iteration. Consequently, at time interval t, only the delay information of the aircraft whose actual arrival time falls within the time window $[t, t+\tau]$ is considered. Aircraft delay information (both arrivals and departures) is input into the model as streams without consideration of unknown future aircraft delays. Algorithm 1 illustrates the proposed rolling horizon approach. The optimization horizon T is rolled forward per Δt time period. A detailed description of this dynamic process can be found in Section 3.2.1.

3.3.3.2 Tabu search heuristic for solving online gate assignment

At each time interval t within the rolling horizon framework, solving the single gate assignment problem proposed in Section 3.2.2 for large-scale instances is computationally

Algorithm 1 Rolling horizon approach

Input: current candidate MP solution v^*, random information $\xi(\omega)$, whole time period T, time window τ, time step Δt, $\mathbf{F}$, $\mathbf{G}$

Output: $(z_t)_{t\in T}$ gate assignment plan for each time interval t, the cost of the assignment at time interval T, $Q^{\omega}(v^*)$

```
Initilize t=0, z_{i,g}^{b,t}=1
while t ≤ T do
    Update â_i^t(ω) and d̂_i^t(ω)
    Solve SP for the optimal assignment for each aircraft i ∈ F at time interval t
    Compute the cost of the assignment at time interval t by Tabu search algorithm
    if a_i^t ≤ t + τ then
        Fix the assignments of the aircraft i ∈ F
        z_{i,g}^{b,t+Δt} ← z_{i,g}^{e,t}
    end if
    if t + Δt ≤ T then
        t ← t + Δt
    else
        t ← T
    end if
end while
```

challenging. Using a MIP solver, such as GUROBI, becomes impractical for direct application, as shown by the numerical results in Section 3.3.3.2. To quickly find high-quality solutions, we adopt a tabu search algorithm. For details on tabu search, see (Glover & Laguna, 1998) and the related literature. Here, we provide an overview of the implementation details for the proposed tabu search algorithm.

We initialize the search by constructing a feasible solution for each subproblem, prioritizing aircraft by arrival time and turnaround time. Gates are assigned to aircraft sequentially following this priority order, ensuring that all available gates are utilized to their maximum capacity, except for the temporary apron (Gate 0). This method guarantees that the initial solution satisfies constraints (3.23)-(3.34).

The search navigates the solution space using three standard neighborhood moves—insert, swap, and interval exchange—commonly used in prior work (Jiefeng & Bailey, 2001; Şeker & Noyan, 2012; Kim et al., 2017) and illustrated in Figure 3.2. An insert move reassigns an aircraft from its assigned gate to a new eligible gate; a swap move exchanges the gates of two aircraft, provided each can feasibly occupy the other's gate; and an interval-exchange move swaps two consecutive aircraft intervals between two gates, contingent on feasibility for all aircraft involved. For each move type, we implement functions that generate candidate moves and apply them: the insert function enumerates all gate reassignments excluding the current gate; the swap function identifies cross-gate aircraft pairs and tests mutual feasibility; and the interval-exchange function finds consecutive intervals across gates and tests whether a full exchange is feasible. Each iteration uses the insert generator, while every third iteration augments the neighborhood by jointly invoking the insert, swap, and interval-exchange generators.

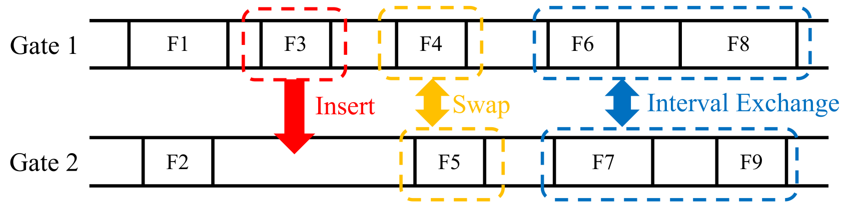


Figure 3.2: Three types of neighborhood moves

To enhance the search and prevent cycling, we maintain a tabu list of temporarily forbidden moves, overridden only by an aspiration criterion. The list is updated dynamically, and its size is tuned to the problem scale: for an instance with n aircraft, we set the tabu tenure to $5+\frac{n}{10}$, following parameterization inspired by Şeker & Noyan (2012). Under the aspiration criterion, a tabu move becomes admissible only if it yields a strictly improved objective over the current best.

The algorithm stops when any of the following holds: (1) a maximum of $600+6n$ iterations is reached; (2) there are $200+2n$ consecutive iterations without improvement; or (3) 600 seconds of wall-clock time elapse, ensuring results are returned within the rolling horizon window.

3.3.4 Accelerating convergence with neighborhood constraints

The proposed decomposition algorithm exhibits diminishing improvement in convergence, with solution quality gains decreasing over successive iterations. To mitigate this issue, we adopt the neighborhood acceleration strategy introduced by Wang & Jacquillat (2020). This strategy involves treating the displacement of aircraft arrivals or departures as if they occurred in earlier iterations. This effectively shrinks the master problem's feasible region and accelerates convergence.

We define a set $V^k \subset F$ for aircraft displaced before iteration k. After several iterations, $V^k \setminus V^{k-1}$ becomes negligible, meaning the displacement of aircraft arrivals or departures has already occurred in earlier iterations. We therefore establish a threshold ε_2, expressed in $\hat{k}$, such that $\frac{UB-LB}{LB} \leq \varepsilon_2$. From the $\hat{k}+1$ iteration onward, equations (3.50) and (3.51) are incorporated into the MP. This ensures that the arrival or departure of any aircraft not displaced before iteration k will remain unchanged in later iterations, effectively constraining the feasible solution space without affecting the validity of optimality cuts (3.48) and (3.49).

$$\sum_{l\in L}\sum_{i\in F}\sum_{t\in T} x_{i,t}^{l}\psi_{i,t}^{Arr} = 0 \quad \forall i \in F \backslash V^{\hat{k}} \tag{3.50}$$

$$\sum_{l\in L}\sum_{i\in F}\sum_{t\in T} y_{i,t}^{l}\psi_{i,t}^{Dep} = 0 \quad \forall i \in F \backslash V^{\hat{k}} \tag{3.51}$$

3.3.5 Full algorithm

In the proposed algorithm, the 2SSIP is decomposed into a MP and several SPs, employing a rolling horizon approach to solve the subproblem. Initially, we establish the iterations, maximum iteration count, lower and upper bounds, convergence threshold, and neighborhood constraint indicators. We compute the initial lower bound $\mathbf{L}$ for $Q(\nu)$.

The algorithm iteratively generates new slot allocations and gate assignment plans, continuously adding optimality cuts and updating the bounds to refine the solution. Neighborhood constraints are introduced to accelerate convergence. The process concludes when the convergence gap reaches ε_1 or the maximum number of iterations is achieved.

Algorithm 3.3.5 synthesizes our solution approach for the two-stage stochastic integer programming. Detailed information on optimality cuts is provided in Section 3.3.2. The rolling horizon framework is discussed in Section 3.3.3. The application of neighborhood constraints is outlined in Section 3.3.4.

Algorithm 2 Full algorithm

Input: coordination parameters u, δ, E, sets F, T, C, S
Output: MP solution v^*, SP solutions $(z^*_\omega)_{\omega \in \Omega}$
Initilize iteration $k = 0$, maximum iteration $k^* = 500$, low bound $LB = -\infty$, upper bound $UB = \infty$, convergence thresholds $\varepsilon_1 = 0.01$, $\varepsilon_2 = 0.05$, convergence gap $GAP = \frac{UB-LB}{LB}$, neighborhood constraints indicator $n = 0$, $n^* = 50$
Compute the lower bound $\mathbf{L}$ of $Q(v)$
while $GAP^k \geq \varepsilon_1$ or $k \leq k^*$ **do**
    Solve MP, denote MP optimal solution by v^*
    Update LB with the MP objective
    for each $\omega \in \Omega$ **do**
        Solve SP by Algorithm 1, denote the optimal solution as z^*_ω
    end for
    Compute $Q(v^*)$
    $UB \leftarrow \min\{UB, c^T v^* + Q(v^*)\}$
    $k \leftarrow k+1$
    $GAP^k \leftarrow \frac{UB-LB}{LB}$
    if $GAP^k == GAP^{k-1}$ **then**
        $n \leftarrow n+1$
    else
        $n \leftarrow 0$
    end if
    if $GAP^k \geq \varepsilon_1$ **then**
        Add optimality cut (3.48) to MP
        for each $\omega \in \Omega$ **do**
            Solve OFF-$\overline{\text{SP}}$, compute $Q^{\omega}_{LP}(v^*)$
        end for
        Add dual offline cut (3.49) to MP
        if $GAP^k \leq \varepsilon_2$ or $n \geq n^*$ **then**
            Add neighborhood constraints (3.50) and (3.51) to MP
        end if
    end if
end while

3.4 Numerical studies

The section assesses the efficiency of the model and solution approach across small to large problem sizes, comparing results against those obtained using GUROBI 11. Real-world scenarios from Shanghai Hongqiao International Airport (SHA) are employed for this case study. All experiments were run in Python on a workstation with a 26-core 2.1 GHz Intel Xeon CPU and 128 GB of RAM.

For this section, Section 3.4.1 summarizes the case study settings including the input data sources, the weight factor ρ, and related parameter settings. Building on this, Section 3.4.2 examines how different SAA sample sizes impact solution quality and identifies a reasonable sample size for subsequent analyses. Section 3.4.3 then compares the performance of solving a single online gate assignment problem using GUROBI and the tabu search heuristic, demonstrating the necessity of the latter in this context. Following this, Section 3.4.4 introduces the instances and model sizes categorized into small, medium, and large airports, which are used throughout the study. To evaluate the algorithm, Section 3.4.5 discusses its convergence properties, while Section 3.4.6 analyzes the computational performance improvements achieved by applying neighborhood constraints. Further, Section 3.4.7 highlights the value of the stochastic solution. Section 3.4.8 compares the computational time and cost of the proposed algorithm, the online formulation, and the offline formulation under two different RHA configurations. Finally, Section 3.4.9 explores the benefits of integrating scheduling and operations, emphasizing the practical implications of the proposed approach.

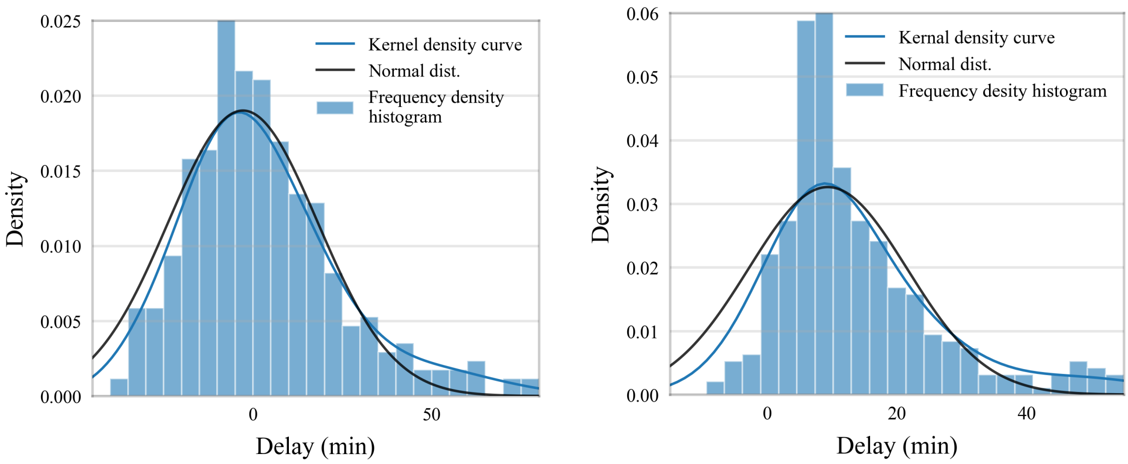
3.4.1 Case study setting

This numerical study examines slot allocation at Shanghai Hongqiao Airport (SHA) during the summer of 2019. Due to data limitation in terms of real slot requests. The displacement indicates deviations from the original schedule without fully considering operational limitations. Additionally, since the planned flight schedules lack priority categorizations for time slots, we assumed that all slots are treated equally. We acknowledge that this assumption may impact the schedule's outcome.

Furthermore, the proposed model considers only the required slot requests for a specific day within the season. Consequently, the scheduling impact over the entire season (e.g., peak day, week, or month) may be deterministic. It is important to note that most slot requests at coordinated airports in China encompass the entire season due to the limited availability of slots. At Shanghai Hongqiao International Airport (SHA), each slot request typically spans the entire season, with airlines often requesting a series of flight time slots to secure scarce resources. Airlines may return unused slots to meet utilization requirements while retaining their historical rights. Although this practice reduces allocative efficiency and contributes to capacity misallocation, it complies with relevant regulations and is therefore expected.

We divide the day into 288 5-minute intervals. The maximum displacement is $\Psi = 60$ minutes; the minimum turnaround time is 65 minutes. A curfew applies from 00:00 to 06:00. First-stage weight factors vary as $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5) = (0.8, 0.6, 0.5, 0.4, 0.2)$. Penalty costs are $\zeta = 50$ and $\sigma = 288$.

For uncertain flight delays, Figure 3.3 shows the annual delay distribution for arrivals and departures at SHA. We fitted curves using kernel density estimation and a normal distribution. The KDE closely matches the normal fit, supporting the assumption that delays are normally distributed. Based on all 2019 historical data, we then generated aircraft-specific arrival and departure delay distributions for each $i \in F$.



(a) Arrival flight delay distribution (b) Departure flight delay distribution

Figure 3.3: Distribution of a single flight's arrival and departure delays at SHA (2019).

3.4.2 Analyzing the sample size N for SAA

As described in Section 3.3, we use SAA to approximate $Q(v)$. For the numerical experiments, we consider two instances with sizes $(F, G) = (450, 150)$ and $(200, 150)$, representing a highly congested and an uncongested airport, respectively; both assume a single terminal ($L = 1$). To approximate the stochastic solution, the sample size N (number of scenarios) is varied from 10 to 500. The objective value obtained with the largest sample size, $N = 500$, serves as the benchmark for solution quality. We employ the problem using the full Algorithm 2, setting the rolling horizon framework configuration with $\Delta t = T$ and $\tau = T$. While this choice substantially reduces computational cost, it may diverge from operational practice. Nonetheless, the primary aim here is to identify an appropriate sample size N for SAA rather than to enforce operational realism.

Table 3.2 provides a detail of the objective function values for various sample sizes N, along with their corresponding 90% confidence intervals and relative errors. We calculate the relative error (RE) using $\text{RE} = \frac{|Z_N - Z_{\text{baseline}}|}{Z_{\text{baseline}}} \times 100\%$, with Z_N is the objective function value for sample size N, and Z_{baseline} denotes the objective function value for the baseline sample size (e.g., $N = 500$). The relative error quantifies the deviation of the objective function value at each sample size from the baseline value, expressed as a percentage of the baseline value. Figure 3.4 illustrates the variation in objective function values across the full range of tested sample sizes N.

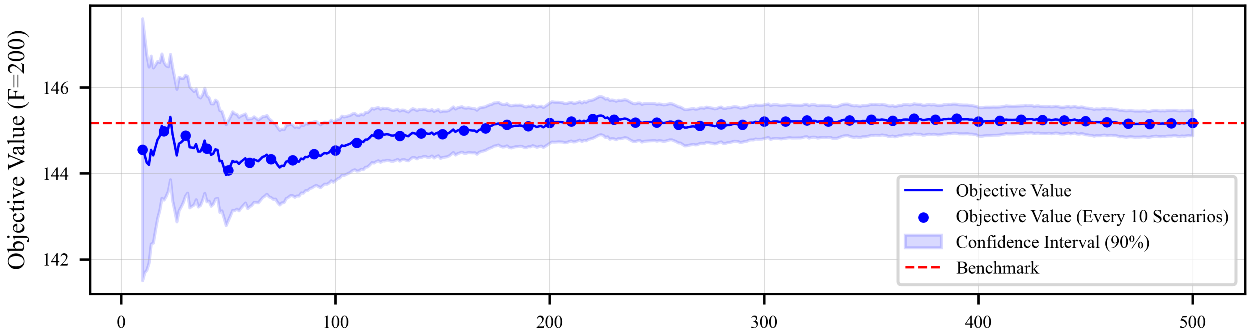
As shown in Table 3.2 and Figure 3.4, for a given F, larger sample sizes N yield narrower confidence intervals. Specifically, when $F = 200$ and $N = 200$, the length of all confidence intervals becomes less than 1 as N increases. Similarly, when $F = 450$ and $N = 80$, the relative errors for all scenarios drop below 0.1% as N increases. Figures (3.4a) and (3.4b) further illustrate that for different values of F, the objective value stabilizes after $N = 200$. Based on these observations, we set $N = 200$ as the sample size for the solution algorithm for the remainder of this study, in order to ensure calculation accuracy while maintaining a reasonable computational effort.

3.4.3 Tabu search heuristics

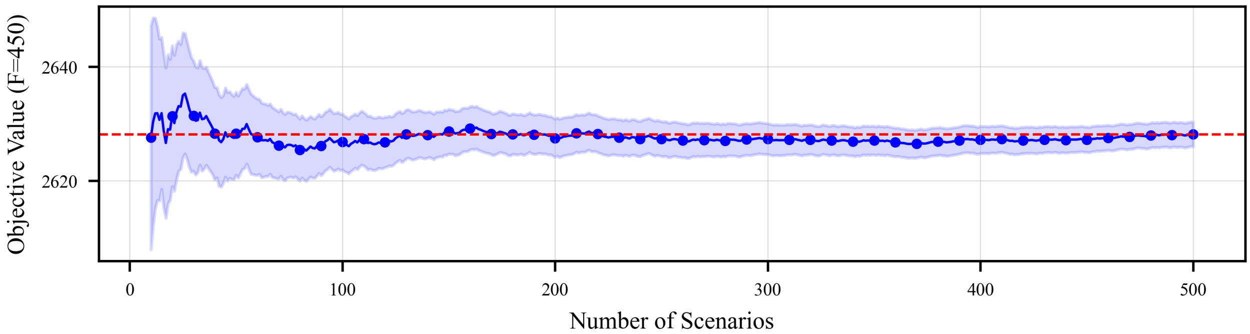
The proposed online gate assignment model yields a large-scale MIP that standard solvers such as GUROBI struggle to solve efficiently within reasonable CPU time. To address this, we benchmark GUROBI against a tabu search heuristic, highlighting the heuristic's advantages in solution quality and computational efficiency.

Table 3.2: Partial objective values, 90% confidence intervals, and relative errors for various sample sizes N ($\rho = 0.5, F = 200, 450$)

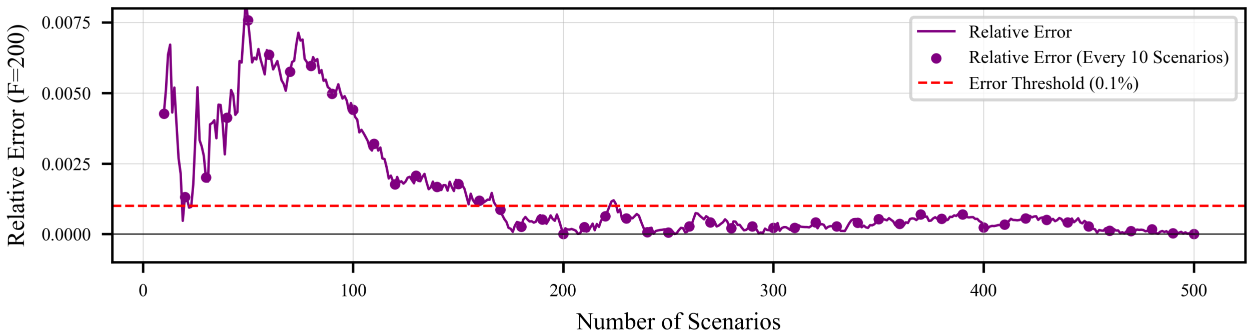
F	N	Objective Value	90% Confidence Interval			RE (%)	Time (min)
			Min	Max	Interval Length		
200	10	145.4	143.1	147.7	4.6	0.08	83.4
	20	145.7	144.2	147.1	3.0	0.28	81.0
	30	145.8	144.4	147.1	2.7	0.35	80.7
	40	145.1	143.9	146.4	2.5	0.10	84.2
	50	144.6	143.4	145.8	2.3	0.45	82.6
	60	144.6	143.6	145.7	2.0	0.42	89.0
	80	144.8	143.9	145.6	1.7	0.32	141.3
	100	145.0	144.2	145.7	1.5	0.21	123.3
	150	145.5	144.9	146.0	1.1	0.14	151.1
	200	145.5	145.1	146.0	0.9	0.19	186.5
	250	145.5	145.1	145.9	0.9	0.19	365.9
	300	145.4	145.0	145.8	0.8	0.11	430.5
	350	145.4	145.1	145.8	0.7	0.13	531.6
	400	145.4	145.0	145.7	0.7	0.08	582.8
	500	145.3	145.0	145.6	0.6	0.00	665.4
450	10	2627.6	2608.0	2647.2	39.2	0.02	420.8
	20	2631.3	2619.0	2643.7	24.7	0.12	425.8
	30	2631.4	2621.7	2641.0	19.3	0.12	429.1
	40	2628.3	2620.2	2636.4	16.2	0.01	437.4
	50	2628.3	2621.0	2635.5	14.5	0.00	428.8
	60	2627.7	2621.0	2634.3	13.3	0.02	444.5
	80	2625.4	2620.0	2630.8	10.8	0.10	630.2
	100	2626.8	2622.0	2631.7	9.7	0.05	617.2
	150	2628.6	2624.7	2632.5	7.8	0.02	846.7
	200	2627.5	2624.1	2630.8	6.7	0.03	833.6
	250	2627.3	2624.3	2630.4	6.1	0.03	1689.3
	300	2627.3	2624.6	2630.0	5.4	0.03	2146.8
	350	2627.1	2624.5	2629.6	5.0	0.04	2570.5
	400	2627.2	2624.8	2629.5	4.7	0.04	2843.9
	500	2628.1	2626.0	2630.3	4.3	0.00	3233.5



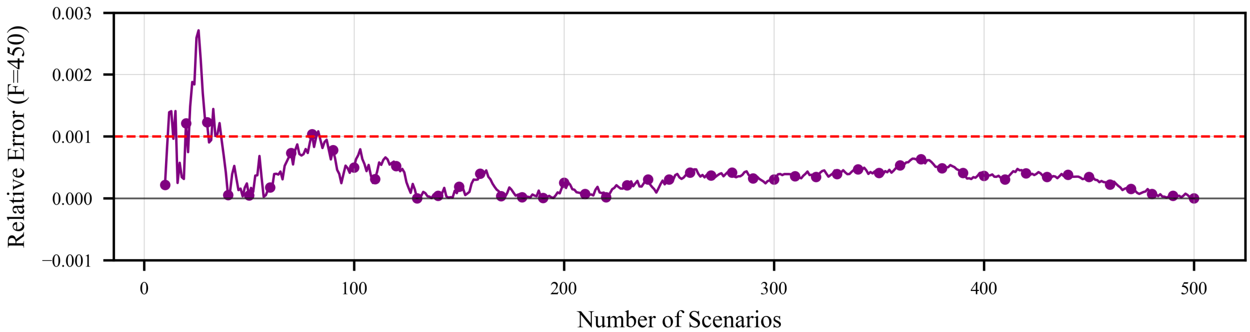
(a) Objective value versus sample size N with 90% confidence intervals (F = 200)



(b) Objective value versus sample size N with 90% confidence intervals (F = 450)



(c) Relative error for different Sample Sizes N (F=200)



(d) Relative error for different Sample Sizes N (F=450)

Figure 3.4: Objective value and relative error for different Sample Sizes N

We generate 8 test instances, each identified as Fx_Gz, where x and z denote the number of aircraft and gate, respectively. The rolling time horizon method is configured with $\Delta t = 1440\,\text{min}$ and $\tau = 10\,\text{min}$. For each test instance, 200 random scenarios with delays are simulated. After solving all 200 scenarios, the average performance metrics for each instance are computed.

To enable GUROBI to solve each online gate-assignment model at time interval t, we set a 1200 seconds time limit and a 0.1% MIP gap. Let Obj_{grb} be the best lower bound reported by GUROBI at the time limit, and Obj_{tbs} be the best objective found by the tabu search under the same limits. We define the optimality gap as:

$$\text{Gap}_{\text{opt}} = \frac{\text{Obj}_{\text{tbs}} - \text{Obj}_{\text{grb}}}{\text{Obj}_{\text{grb}}}.$$

As shown in Table 3.3, for smaller instances, the tabu search heuristic achieves solutions of comparable quality to those obtained by GUROBI, but with a substantial reduction in computational time. It is worth mentioning that GUROBI's computation time increases significantly when dealing with congested airports. For example, for the F50_G10 instance, GUROBI consistently reaches the maximum time limit before solving the online gate model to optimality. Conversely, the heuristic attains a high-quality solution quickly. For problem instances with $F = 450$, solver fails to return any feasible solutions within the specified time and memory constraints. This result highlights the efficacy of the heuristic approach, particularly for large-scale and computationally intensive problem instances. Therefore, for the reminder of this study, Tabu Search heuristic algorithm is utilised as the solution approach for solving sub problems.

3.4.4 Test instances

This numerical study employs three sets of examples to simulate small, medium, and large airports. The total of 6 test instances are produced based on the 2019 summer schedule of Shanghai Hongqiao International Airport, with two for each airport size. The number of aircraft ($|F|$), time intervals ($|T|$), terminals ($|L|$), gates ($|G|$), scenarios ($|\Omega|$), and the cor-

Table 3.3: Comparison of GUROBI and Tabu Search Performance

Instance	GUROBI		Tabu Search		Gap
	Obj	CPU Time (s)	Obj	CPU Time (s)	
F10_G5	9.2	2.9	9.2	51.2	0.00%
F30_G5	477.9	203.8	478.2	70.1	0.06%
F50_G10	180.8	15200.6	181.6	125.0	0.40%
F100_G25	38.5	13433.0	38.7	338.3	0.52%
F150_G50	42.8	4488.5	42.8	725.3	0.00%
F200_G100	72.8	6727.7	74.9	1358.9	2.94%
F300_G100	84.3	14793.8	89.3	2985.9	5.90%
F450_G150	-	-	138.0	6782.3	-

responding problem sizes for each test instance are summarized in Table 3.4. The declared capacities of these airports are detailed in Table 3.5. For different terminals, we assume that different terminals have equal apron capacity.

Table 3.4: Test instances and related model sizes

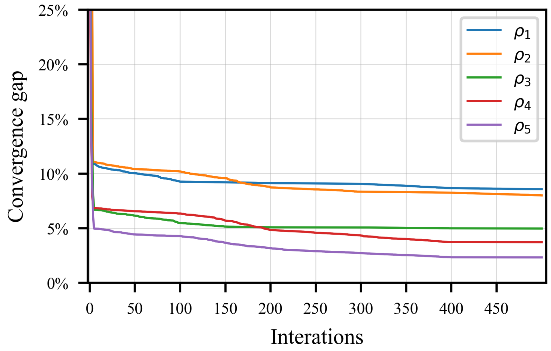
Instances	$\|F\|$	$\|T\|$	$\|L\|$	$\|G\|$	$\|\Omega\|$	Variables	Constraints
Case 1	50	288	1	15	200	369,426	8,011,892
Case 2	50	288	2	15	200	398,802	8,013,196
Case 3	200	288	1	50	200	4,275,976	406,243,242
Case 4	200	288	2	50	200	4,391,752	406,244,996
Case 5	450	288	1	150	200	27,620,226	6,116,045,492
Case 6	450	288	2	150	200	27,880,002	6,116,047,996

Table 3.5: Parameter settings for different cases

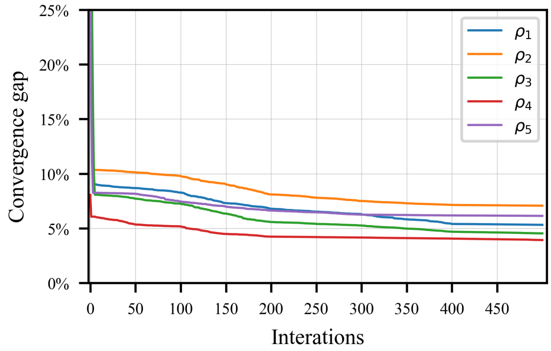
Parameter (notation)	Description	Case 1-2	Case 3-4	Case 5-6
Runway Capacity ($u_{t,c}^{Arr(Dep)}$)	Arrivals	4 per 60 min	15 per 60 min	36 per 60 min
	Departures	4 per 60 min	15 per 60 min	36 per 60 min
	Total	6 per 60 min	22 per 60 min	48 per 60 min
Passenger Terminal Capacity ($E_{t,c}^{Arr(Dep)}$)	Arrivals	900 per 60 min	3600 per 60 min	9000 per 60 min
	Departures	900 per 60 min	3600 per 60 min	9000 per 60 min
Apron Capacity ($\sum_{l\in L} \delta_{t,s}^{l}$)	Passenger	14	49	148
	Cargo	1	1	2
Turnaround Time ($T_{\min,i}/T_{\max,i}$)	Minimum/Maximum	65 min / 180 min	65 min / 180 min	65 min / 180 min
Maximum Displacement (Ψ)	-	60 min	60 min	60 min
Load Factor ($\pi_i^{Arr(Dep)}$)	-	0.8	0.8	0.8
Gates (g)	Passenger	14	49	148
	Cargo	1	1	2
Parking Interval (η)	Minimum	20 min	20 min	20 min

3.4.5 Convergence of proposed algorithm

We evaluate the algorithm's performance with the optimal cut and dual offline cut, as introduced in Section 3.3.2. To ensure an unbiased evaluation, neighborhood constraint acceleration is not employed. Figure 3.5 illustrates the convergence gap ($\frac{UB-LB}{LB}$) over 500 iterations for Case 5 and Case 6 with different ρ.



(a) Convergence gap with different ρ for Case 5



(b) Convergence gap with different ρ for Case 6

Figure 3.5: Convergence process for Case 5 and Case 6

Table 3.6 shows that all instances exhibit similar convergence results. Moreover, we observed that the rate of convergence improvement diminished as the number of iterations increased, motivating the application of neighborhood constraints to accelerate convergence. The performance of using neighborhood constraints is further analyzed in Section 3.4.6.

Table 3.6: Convergence gap for all instances with different ρ

	Case 1					Case 2					Case 3				
Iterations	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5
5	11.05%	11.17%	11.00%	6.56%	5.72%	12.02%	11.73%	11.02%	6.52%	5.44%	7.22%	6.83%	6.57%	5.44%	4.67%
50	6.71%	10.12%	9.86%	6.40%	5.07%	9.60%	11.11%	9.39%	5.30%	4.80%	6.98%	6.45%	6.01%	5.01%	4.07%
100	5.66%	10.12%	9.57%	5.36%	5.04%	9.08%	10.98%	9.20%	5.23%	3.94%	6.79%	6.09%	5.93%	4.96%	4.04%
150	4.23%	9.84%	9.26%	5.28%	4.94%	8.94%	10.43%	9.07%	5.20%	3.86%	6.62%	6.09%	5.90%	4.96%	3.90%
200	4.11%	9.83%	9.09%	5.20%	4.94%	8.85%	10.33%	8.86%	5.08%	3.78%	6.62%	6.00%	5.90%	4.63%	3.61%
300	3.94%	9.69%	8.95%	5.13%	4.50%	8.56%	10.17%	8.70%	5.02%	3.52%	6.51%	5.91%	5.90%	4.60%	3.59%
400	3.83%	9.67%	8.86%	5.07%	4.32%	8.37%	10.05%	8.57%	4.98%	3.44%	6.31%	5.86%	5.85%	4.55%	3.55%
500	3.68%	9.57%	8.80%	5.04%	4.22%	8.24%	10.00%	8.36%	4.92%	3.07%	6.31%	5.86%	5.80%	4.55%	3.55%

	Case 4					Case 5					Case 6				
Iterations	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5
5	7.11%	7.03%	6.73%	5.46%	4.96%	10.85%	11.07%	6.68%	6.83%	4.96%	9.02%	10.36%	8.08%	6.07%	8.25%
50	6.90%	6.50%	6.15%	4.97%	4.42%	10.02%	10.39%	6.14%	6.54%	4.42%	8.68%	10.12%	7.71%	5.35%	8.16%
100	6.45%	6.18%	6.10%	4.94%	4.27%	9.25%	10.18%	5.46%	6.33%	4.27%	8.27%	9.79%	7.26%	5.18%	7.45%
150	6.45%	6.18%	6.10%	4.94%	3.66%	9.19%	9.57%	5.14%	5.69%	3.66%	7.31%	9.02%	6.35%	4.49%	6.98%
200	6.45%	6.18%	5.95%	4.94%	3.16%	9.12%	8.72%	5.07%	4.83%	3.16%	6.79%	8.10%	5.58%	4.24%	6.63%
300	6.45%	6.18%	5.95%	4.87%	2.72%	9.05%	8.33%	5.06%	4.32%	2.72%	6.30%	7.50%	5.26%	4.16%	6.25%
400	6.40%	6.02%	5.89%	4.87%	2.33%	8.66%	8.24%	4.98%	3.72%	2.33%	5.40%	7.14%	4.69%	4.06%	6.18%
500	6.40%	4.17%	5.85%	4.87%	2.32%	8.56%	7.99%	4.96%	3.71%	2.32%	5.32%	7.07%	4.54%	3.93%	6.14%

3.4.6 Computational performance of applying neighborhood constraints

To evaluate the acceleration performance of neighborhood constraints, we establish a baseline using the combined optimal and dual offline cuts, referred to as BS. We then incorporate the neighborhood constraints, corresponding to the full Algorithm 2.

Table 3.7 summarizes results for all instances. It reports the lower bound, upper bound, convergence gap ($Gap = (\mathrm{UB} - \mathrm{LB})/\mathrm{LB}$), number of iterations (k), and total computational time (minutes) at termination.

Table 3.7: Effectiveness of the acceleration methods

Instance	ρ	BS					Algorithm 2				
		LB	*UB*	*Gap*	*k*	*Time* (min)	*LB*	*UB*	*Gap*	*k*	*Time* (min)
Case 1	ρ_1	275.3	285.4	3.68%	500	1004	282.6	282.6	0.00%	65	85
	ρ_2	211.1	231.3	9.57%	500	1022	220.6	220.7	0.02%	125	219
	ρ_3	234.7	255.4	8.80%	500	1019	240.5	241.3	0.33%	186	351
	ρ_4	172.1	180.7	5.04%	500	1009	174.8	174.8	0.00%	264	534
	ρ_5	176.9	184.3	4.22%	500	1049	187.0	188.9	1.00%	422	878
Case 2	ρ_1	295.8	320.1	8.24%	500	1070	297.9	297.9	0.00%	88	137
	ρ_2	261.1	287.3	10.00%	500	1025	267.9	269.8	0.68%	256	524
	ρ_3	244.4	264.8	8.36%	500	993	253.0	253.0	0.00%	266	520
	ρ_4	228.7	240.0	4.92%	500	1066	238.0	238.8	0.33%	258	501
	ρ_5	204.7	211.0	3.07%	500	1025	208.1	209.3	0.58%	386	766
Case 3	ρ_1	1070.5	1138.0	6.31%	500	11113	1081.0	1081.1	0.01%	322	6647
	ρ_2	996.1	1054.5	5.86%	500	10558	1009.5	1016.0	0.64%	355	7203
	ρ_3	905.4	958.0	5.80%	500	10883	959.3	990.1	3.21%	500	11198
	ρ_4	852.1	890.9	4.55%	500	11036	893.3	925.0	3.55%	500	10592
	ρ_5	660.4	683.8	3.55%	500	11072	737.4	759.4	2.99%	500	11335
Case 4	ρ_1	1013.5	1079.4	6.50%	500	11188	1096.3	1098.8	0.23%	371	7741
	ρ_2	952.5	1012.0	6.25%	500	11254	1011.7	1018.7	0.69%	366	8353
	ρ_3	917.6	972.5	5.99%	500	11282	960.7	965.6	0.51%	488	10919
	ρ_4	890.3	933.4	4.84%	500	11086	899.7	919.7	2.22%	500	10654
	ρ_5	688.3	707.9	2.83%	500	10801	744.1	760.9	2.26%	500	10572
Case 5	ρ_1	2616.2	2840.2	8.56%	500	52203	2628.7	2643.4	0.56%	462	48677
	ρ_2	2568.8	2774.1	7.99%	500	52638	2639.9	2642.8	0.11%	496	52117
	ρ_3	2509.7	2634.1	4.96%	500	52334	2519.9	2554.2	1.36%	500	52199
	ρ_4	2396.3	2485.1	3.71%	500	52290	2486.0	2517.1	1.25%	500	52245
	ρ_5	2340.7	2395.0	2.32%	500	53048	2387.1	2440.1	2.22%	500	52699
Case 6	ρ_1	2679.6	2822.1	5.32%	500	52442	2725.6	2732.3	0.24%	489	51792
	ρ_2	2572.7	2754.6	7.07%	500	52286	2662.2	2704.4	1.58%	500	52981
	ρ_3	2590.6	2708.2	4.54%	500	53260	2620.0	2682.6	2.39%	500	52550
	ρ_4	2515.2	2614.0	3.93%	500	52783	2564.5	2607.7	1.69%	500	52558
	ρ_5	2416.6	2565.1	6.14%	500	53135	2452.5	2529.2	3.13%	500	52801
Average		1215.3	1282.8	5.76%	500	21566	1248.3	1265.5	1.13%	389	20678

The comparison between the baseline solution (BS) and Algorithm 2 demonstrates significant improvements attributable to the inclusion of neighborhood constraints. These constraints reduce the average *Gap* from 5.76% to 1.13%, with more pronounced benefits observed when both algorithms terminate after the same number of iterations, as seen in Case 6. Additionally, neighborhood constraints enhance performance by reducing the average CPU time from 21,566 minutes to 20,678 minutes, achieved by narrowing the solution space of the MP in subsequent iterations.

Moreover, neighborhood constraints improve the relevance of optimality cuts to first-stage solutions, such as potentially rejected slot requests, effectively lowering the average convergence gap at termination. This neighborhood-based approach thus accelerates the decomposition algorithm for two-stage stochastic procedures. However, while these constraints expedite convergence, they can result in increased total costs, as the average lower bound rises from 1215.3 to 1248.3.

When comparing cases with the same number of aircraft but differing numbers of terminals (e.g., Case 5 and Case 6), we observe that cases with more terminals incur higher costs. This is because additional terminals impose stricter constraints, such as constraints (3.12), (3.13), (3.14), (3.17) and (3.18). Moreover, reducing ρ increases the weight on the second-stage cost, which slows convergence across all cases.

3.4.7 The value of stochastic solution

As a key metric in stochastic optimization, the value of stochastic solution (VSS) quantifies the benefit of using a randomized programming approach by comparing expected costs under uncertainty to those from a deterministic model. The calculations of the expected value of the solution (EEV) and VSS are detailed in Birge & Louveaux (2011). Table 3.8 reports the values of the objective value from algorithm 2, EEV, and VSS for Case 5 and case 6 under different ρ.

The VSS values range from 3.58% to 18.66% of the objective value, highlighting the substantial improvements achievable through the stochastic optimization approach compared to the deterministic method in the first-stage airport slot allocation. This is because

Table 3.8: Value of Stochastic Solutions Information

Instance	ρ	Obj	EEV	VSS	$\frac{VSS}{Obj}$
Case 5	ρ_1	2631.9	2726.2	94.3	3.58%
	ρ_2	2556.6	2734.3	177.7	6.95%
	ρ_3	2574.8	2815.0	240.2	9.33%
	ρ_4	2473.4	2781.3	307.9	12.45%
	ρ_5	2378.7	2740.5	361.9	15.21%
Case 6	ρ_1	2725.6	2875.2	149.6	5.49%
	ρ_2	2662.2	2828.3	166.1	6.24%
	ρ_3	2620.0	2878.2	258.2	9.86%
	ρ_4	2564.4	2904.2	339.8	13.25%
	ρ_5	2452.5	2910.2	457.7	18.66%

the cost of airport slot allocation is highly sensitive to the optimal balance between slot assignment costs and ground operation delays under stochastic operating conditions.

As ρ decreases, the ratio of $\frac{VSS}{Obj}$ increases, indicating that assigning greater weight to the second-stage costs amplifies the impact of second-stage uncertainty on first-stage decision-making. These findings demonstrate that a larger ρ, which results in smaller displacement, reduces the benefits of stochastic optimization.

3.4.8 Comparison of rolling horizon approach configurations

The section compares computational time and objective cost across the proposed algorithm, the online formulation, and the offline formulation, under two rolling horizon (RHA) configurations. To evaluate computational performance and solution quality, we selected Case 6 with a maximum of 500 iterations. The first configuration represents the online configuration RHA, while the second configuration, SHA, serves as the offline benchmark.

(1) RHA ($\Delta t = 10mins$, $\tau = 120mins$): The configuration rolls forward every 10 minutes, revealing uncertain information about aircraft arriving within the time window [t, $t+\tau$].

(2) SHA ($\Delta t = T$, $\tau = T$): The offline problem is solved with complete knowledge of aircraft delays.

The variation of the total cost and the proposed algorithm's computational performance for the two tested configurations are shown in Table 3.9. The RHA demonstrates poorer computational performance and higher costs compared to the SHA. However, its primary advantage lies in its ability to take the dynamic of real-world operations. This is because it treats the second-stage gate assignment problem as a real-time problem, closely reflecting real-world scenarios. The online model imposes stricter constraints than the offline model, as it requires continuous decision-making and real-time updates, both of which significantly influence gate assignment decisions. While the rolling horizon approach demands greater computational effort due to these dynamic adjustments, it better represents the operational complexities of airport management. In contrast, the offline model is computationally more efficient but fails to fully replicate the flexibility and responsiveness required in practice.

Table 3.9: Comparison of results for RHA/SHA configuration for Case 5

	RHA					SHA				
ρ	*LB*	*UB*	*Gap*	*k*	*Time* (min)	*LB*	*UB*	*Gap*	*k*	*Time* (min)
ρ_1	2628.7	2643.4	0.56%	462	48677	2577.6	2577.6	0.00%	65	232
ρ_2	2639.9	2642.8	0.11%	496	52117	2627.5	2627.5	0.00%	254	834
ρ_3	2519.9	2554.2	1.36%	500	52199	2498.2	2505.2	0.28%	265	852
ρ_4	2486.0	2517.1	1.25%	500	52245	2388.2	2401.4	0.56%	168	492
ρ_5	2387.1	2440.1	2.22%	500	52699	2326.9	2347.4	0.88%	305	906

3.4.9 Benefits of scheduling and operations integration

Figure 3.6 illustrates the trade-offs between the number of rejected slot requests, the number of displaced slot requests, the total displacement, and the expected cost in the gate assignment problem by varying ρ in the problem of joint ASA and gate assignment. These results are compared with a sequential approach, which first applies the airport slot allocation model (Eqs. (3.52a)-(3.52b)) and then uses the resulting flight schedule to execute the online gate assignment model (Equations (3.22)-(3.34)).

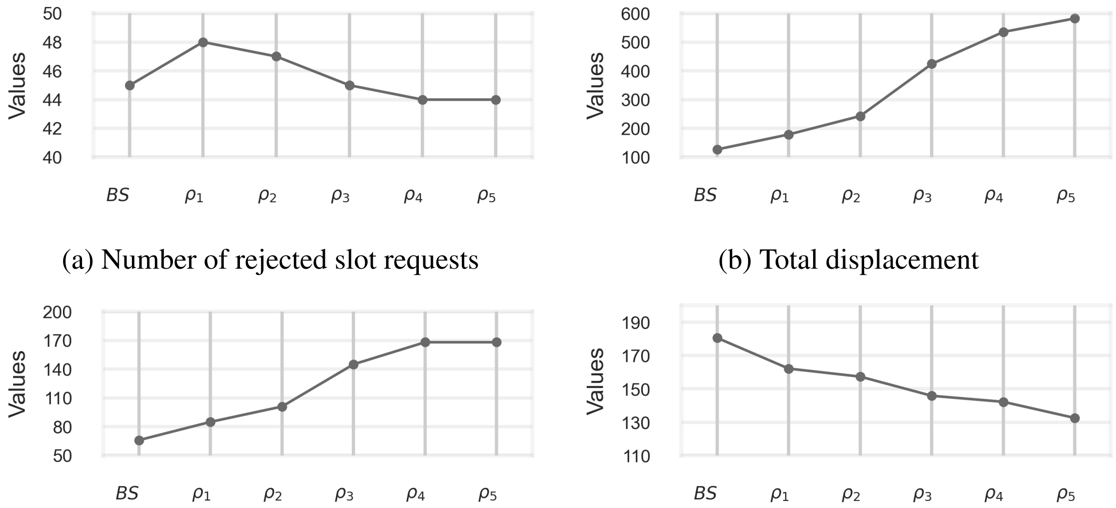
$$\min \zeta \left(|F| - \sum_{i \in F} \mu_i \right) + \sum_{i \in F} \left[\sum_{t \in T} x_{i,t} \psi_{t,a} + \sum_{t \in T} y_{i,t} \psi_{t,d} \right] \tag{3.52a}$$

$$\text{s.t.} Eqs.(3.2) - (3.21) \tag{3.52b}$$

To make the benefit of integration quantitatively explicit, we use the expected second-stage cost (Figure 3.6d) as an operational performance indicator, since it directly measures downstream gate-management consequences (i.e., ground-delay-related gate waiting and the use of the temporary apron) under uncertainty. Under the sequential baseline (BS), the first-stage schedule is optimized without internalizing this operational cost, and the resulting schedule is then evaluated by the online gate assignment model. In contrast, the joint two-stage model internalizes this operational impact during planning, so first-stage adjustments (e.g., accepting/rejecting requests and displacing slots) are made only when they can translate into expected savings in the operational stage. The systematic gap observed in Figure 3.6d—where BS yields a higher expected second-stage cost than the joint model across different values of ρ—provides direct quantitative evidence that decoupled planning and operations can lead to measurable inefficiencies in gate management.

The results for Case 5 demonstrate the benefits of considering the interdependence between planning and operations. Optimizing ASA significantly reduces the expected costs in the gate assignment problem, particularly as the scale of optimization increases. When greater intervention (e.g., ρ_5) is implemented in the airport operational stage, it effectively reduces the expected cost of the second-stage operation. Despite significant gains in rejected slot requests and total displacement, the sequential approach (indeedx as BS) leads to higher gate assignment costs compared with jointly optimizing ASA and gate assignments in two stages. The evidence points to the critical role of integrated planning, with planning-phase improvements driving major operational cost savings.

In summary, Figure 3.6 demonstrates that first-stage planning optimization results in significant second-stage cost reductions, emphasizing the value of integrated planning in airport operations.



(a) Number of rejected slot requests

(b) Total displacement

(c) Number of displaced slot

(d) Expected cost of second-stage

Figure 3.6: Comparison of different metrics under various ρ

3.5 Summary

This study introduces a modeling framework that jointly optimizes planning-level scheduling (airport slot allocation) before uncertainty is resolved and operational-level gate assignment afterward. The problem is formulated as a 2SSIP. Departing from conventional gate-assignment formulations, we cast the second stage as an online model with continuously updated stochastic information, solved via a rolling-horizon scheme. From the dual LP relaxation of the offline gate assignment problem, we derive dual offline cuts that yield tighter global lower bounds than standard optimality cuts in 2SSIP. We further incorporate neighborhood constraints to accelerate convergence of the decomposition algorithm.

We utilized real data from Shanghai Hongqiao Airport for computational illustration. In the largest test instance, which featured 20 million variables and 6 billion constraints, the proposed algorithm delivered superior results within reasonable computation times. The results from the instances demonstrate the convergence of the decomposition algorithm and highlight the benefits of incorporating neighborhood constraints, as opposed to not using acceleration strategies. The case study indicates that this approach can provide valuable decision support for scheduling coordination at airports, effectively capturing the interdependencies between planning and operations. The proposed method can assist airports of various sizes in determining the extent and timing of scheduling adjustments needed to mitigate congestion.

There are multiple avenues to extend this study. Firstly, while this study addresses the slot allocation and gate assignment problems for one airport, there is significant potential to integrate more complex network-wide scheduling and allocation problems. Secondly, robustness is increasingly important. Achieving a more robust ASA decision can be accomplished by considering airport operations optimized for worst-case scenarios, resulting in resilient flight schedules.

Chapter 4

A Two-Stage Robust Optimization for Airport Slot Allocation and Gate Assignment

4.1 Introduction

According to the 2024 Statistical Bulletin on the Development of the China Civil Aviation Industry, passenger and cargo traffic in China's civil aviation have rebounded markedly, indicating a robust recovery from COVID-19. Continued growth will likely intensify demand, heightening congestion risks such as lower punctuality and delays. While weather and airline operations influence schedules, the fundamental constraint is the gap between demand growth and airport capacity (Ribeiro et al., 2019; Bi et al., 2022).

Given the substantial investments required to expand airport capacity, optimizing the use of current resources becomes imperative. Airport slot allocation (ASA) is a strategic long-term planning process aimed at addressing demand-capacity imbalances, particularly in congested airports as defined by the IATA. This process involves collaboration among slot coordinators, airports, and airlines to align flight schedules with airport coordination parameters. Conducted bi-annually, the ASA process sets the flight schedule for the upcoming season, beginning six months in advance and typically remaining unchanged during the operational period. In summary, slot allocation apportions slots to airlines within regulatory and operational constraints to make efficient use of limited airport capacity (Zografos et al., 2017).

Despite careful planning, actual operations often deviate from scheduled plans due to unforeseen disruptions such as mechanical issues or adverse weather. An overloaded schedule exacerbates airport congestion, as delays in initial flights can cascade through the interconnected flight network. For instance, in 2023, approximately 13.2% of flights in China experienced delays, with a significant portion due to meteorological and airline-related factors. Revising flight schedules in response to major disruptions is necessary but can lead to efficiency losses and reduced profitability. Therefore, incorporating operational uncertainties into the ASA decision-making process can enhance overall performance.

Gate assignment represents a critical element of short-term airport management, often subject to uncertainties (Şeker & Noyan, 2012; Kim et al., 2023). Gates facilitate essential ground services for arriving flights, making their effective utilization crucial. Typically, gate assignments are based on planned flight schedules; however, delays frequently disrupt these plans, necessitating real-time reassignments that impact punctuality and passenger satisfaction.

Traditional approaches to airport scheduling often overlook the interconnected nature of planning-level ASA and operational-level gate assignment. Recognizing this interdependence is essential, as gate assignments are influenced by flight schedules derived from ASA. By integrating uncertainties in gate assignments during the ASA process, operational efficiency can be improved, delays minimized, and resource utilization optimized.

In this study, we introduce a two-stage robust optimization framework for airport slot allocation and gate assignment, explicitly considering operational uncertainties. Our approach aims to enhance decision-making by bridging the gap between planning and operations, leading to more resilient airport management solutions.

Our contributions are as follows: (i) We formulate an integrated problem that couples slot allocation and gate assignment while treating aircraft arrivals as uncertain. The goal is a flight schedule that maintains feasibility and quality under operational perturbations. (ii) We develop unified models for airport slot allocation and gate assignment that embed a budgeted uncertainty set for arrival times, producing a two-stage robust counterpart. The first-stage determined each aircraft's arrival and departure. The second-stage uses binary variables for gate choices and continuous variables for gate-occupancy timing. The robust objective minimizes total cost against worst-case uncertainty within the budget. (iii) We develop a nested column-and-constraint generation algorithon (C&CG) to deal with the robust model efficiently. Computational studies on realistic instances demonstrate the effectiveness and scalability of the approach.

4.2 Problem description

4.2.1 Notation and problem statement

This section tackles the joint optimization of two interdependent airport problems: slot allocation at the planning phase and gate assignment at the operational phase. The joint problem takes as inputs: (i) airlines' slot requests, (ii) airport capacity, and (iii) operating scenarios. Preferred schedules derive from airlines' slot requests. Airport capacity includes coordination parameters and the number and types of gates. Coordination parameters specify, for each period, the maximum numbers of arrivals and departures, reflecting runway, apron, and terminal capacities. Scenarios capture flight delays under varying operating conditions.

The joint problem tackles ASA and gate assignment. In the first-stage, ASA schedules each aircraft's arrivals and departures, producing a complete flight timetable. In the second stage, the gate assignment allocates aircraft to gates for ground operations, thereby evaluating the operational impact of each candidate flight schedule on overall airport performance.

In the first stage, ASA aims to minimize the cost of displacement from airlines' preferred schedules while adhering to airport capacity constraints. Slot requests typically specify either an arrival or a departure time; however, this study assumes that each request includes both, executed by the same aircraft, thereby aligning more closely with real-world scenarios.

Unlike ASA, which must contend with operational uncertainty, gate assignment are determined after that uncertainty is revealed. In the second-stage, the gate assignment minimizes the cost of aircraft parking while enforcing non-overlapping gate assignments within each scenario.

In summary, the joint problem of ASA and gate assignment can be described as follows. Initially, ASA allocates arrival and departure times to each aircraft at a single airport, i.e., the first stage of the decision problem. Once these slots are determined, gates are assigned to aircraft in the second stage, taking into account real-time operational uncertainties. To fully capture the interdependencies between these stages, we implement a two-stage robust optimization approach that can jointly optimize airport slot allocation and gate assignment. Our objective is to minimize rejected slot requests, total displacement, and the cost of the second stage across different scenarios (considering flight arrival uncertainty). The detailed model is presented in Section 4.2.2 and Section 4.2.3.

4.2.2 A deterministic model

We first formulate a deterministic MIP model for the integrated airport slot allocation and gate assignment problem. In this setting, all inputs are assumed known and fixed. The notation and input parameters used by the MIP are defined as follows:

Table 4.1: Notation for model formulation

Notation	Description
Sets	
F	Set of aircraft, indexed by i
T	$\{0,\dots,\lvert T\rvert\}$, set of time intervals, indexed by t
C	Set of airport capacity timescales, indexed by c
S	$\{Pax, Cgo\}$, set of service types, indexed by s, which is divided into passenger (Pax) and cargo (Cgo) services.
G	Set of gates, indexed by g
Parameters	

Notation	Description
ζ	Penalty cost for rejecting a slot request
$r_{i,a(d)}$	Requested time slots of aircraft i, categorized as $r_{i,a}$ (arrival) and $r_{i,d}$ (departure)
$T_{\min,i}$, $T_{\max,i}$	Minimum and maximum turnaround time of aircraft i
$\beta_{i,s}^{\text{Arr(Dep)}}$	Equals 1 if aircraft i's arrival (departure) is of service type s, 0 otherwise
$u_{t,c}^{\text{Arr(Dep,Total)}}$	Runway capacity for arrivals (departures, or total) within the time window $[t, t+c)$ for a time interval of length c
$\delta_{t,s}$	Apron capacity for aircraft of service type s at time interval t
$E_{t,c}^{\text{Arr(Dep)}}$	Passenger terminal capacity for arrivals (departures) during time window $[t, t+c)$ for a time interval of length c
e_i	Number of seats requested for aircraft i
$\psi_{i,t}^{\text{Arr(Dep)}}$	Displacement of time slot request for aircraft i: $\psi_{i,t}^{\text{Arr}} = \lvert t - r_{i,a} \rvert$, $\psi_{i,t}^{\text{Dep}} = \lvert t - r_{i,d} \rvert$
Ψ	Maximum displacement of all aircraft
$\pi_i^{\text{Arr(Dep)}}$	Load factor for arrival (departure) of aircraft i
σ	Penalty cost of parking at temporary apron (Gate 0)
$\hat{a}_i^t$	Random parameter: deviation of the real arrival time a_i^t of aircraft i from the scheduled time at interval t under scenario $\omega \in \Omega$
$\hat{d}_i^t$	Random parameter: deviation of the real departure time d_i^t of aircraft i from the scheduled time at interval t under scenario $\omega \in \Omega$
η	Minimum time interval between two successive aircraft parked at the same gate
$o_{i,g}$	Equals 1 if aircraft i can park at gate g, 0 otherwise
Variables and expressions	
$x_{i,t}$	Equals 1 if aircraft i is allocated to arrive at time t, 0 otherwise
$y_{i,t}$	Equals 1 if aircraft i is allocated to depart at time t, 0 otherwise

Notation	Description
μ_i	Equals 1 if aircraft i is allocated both an arrival and departure time slot, 0 otherwise
$n_{t,s}$	Number of service type s aircraft parked at time interval t
$z_{i,g}$	Equals 1 if gate g is assigned to aircraft i, 0 otherwise
a_i	Arrival time of aircraft i
d_i	Departure time of aircraft i
p_i^b	The start time of the aircraft's parking at the gate
p_i^e	The end time of the aircraft's parking at the gate

We refer to the notation and model formulation from the ASA model proposed by Katsigiannis & Zografos (2021) and the gate assignment model proposed by Li et al. (2021). Based on their work, we formulate a deterministic MIP model for the joint ASA and gate assignment problem. Table 4.1 summarizes the notation.

We cast the deterministic model with the following objectives: (i) minimizing the number of rejected slot requests (C_{Sr}), (ii) minimizing the total displacement (C_{Td}), (iii) minimizing the total cost of operational ground delay (C_{Gd}), and (iv) minimizing the cost of aircraft assigned to the temporary apron (C_{Ta}). Thus, we have:

$$C_{Sr} = \zeta \left(|F| - \sum_{i \in F} \mu_i \right) \tag{4.1}$$

$$C_{Td} = \sum_{i \in F} \left[\sum_{t \in T} x_{i,t}\, \psi_{i,t}^{Arr} + \sum_{t \in T} y_{i,t}\, \psi_{i,t}^{Dep} \right] \tag{4.2}$$

$$C_{Gd} = \sum_{i \in F} \left(p_i^b - a_i \right) + \left(p_i^e - d_i \right) \tag{4.3}$$

$$C_{Ta} = \sum_{i \in F} \sigma z_{i,0} \tag{4.4}$$

where C_{Sr} represents the total cost of rejected time slot requests and ζ is the penalty coefficient for rejecting an aircraft i, typically $\zeta = |T|$. C_{Td} denotes the sum of arrival and departure displacements. The total ground-delay cost, C_{Gd}, comprises two terms: the gap between gate-stand entry and the actual arrival time, and the gap between gate-stand exit and the actual departure time. C_{Ta} represents the total cost of an aircraft parking at a temporary apron, with gate 0 indicating the temporary apron (assuming infinite capacity), and σ is the penalty coefficient for parking there.

The deterministic model is described by Eqs. (4.5)–(4.20). The objective function minimizes rejected slot requests, total displacement, ground delays, and the costs associated with aircraft assigned to the temporary apron.

$$\min C_{Sr} + C_{Td} + C_{Gd} + C_{Ta} \tag{4.5}$$

$$\sum_{t \in T} x_{i,t} \leq 1 \quad \forall i \in F \tag{4.6}$$

$$\sum_{t \in T} y_{i,t} \leq 1 \quad \forall i \in F \tag{4.7}$$

$$\sum_{t \in T} x_{i,t} = \sum_{t \in T} y_{i,t} = \mu_i \quad \forall i \in F \tag{4.8}$$

$$\mu_i T_{\min,i} \leq \sum_{t \in T} y_{i,t} t - \sum_{t \in T} x_{i,t} t \leq \mu_i T_{\max,i} \quad \forall i \in F \tag{4.9}$$

$$\sum_{t \in T} x_{i,t} \psi_{i,t}^{Arr} \leq \Psi \quad \forall i \in F \tag{4.10}$$

$$\sum_{t \in T} y_{i,t} \psi_{i,t}^{Dep} \leq \Psi \quad \forall i \in F \tag{4.11}$$

$$\sum_{i \in F} \sum_{t' \in [t, t+c-1]} x_{i,t'} \leq u_{t,c}^{Arr} \quad \forall c \in C, t \in [0, |T| - c] \tag{4.12}$$

$$\sum_{i \in F} \sum_{t' \in [t, t+c-1]} y_{i,t'} \leq u_{t,c}^{Dep} \quad \forall c \in C, t \in [0, |T| - c] \tag{4.13}$$

$$\sum_{i \in F} \sum_{t' \in [t, t+c-1]} \left(x_{i,t'} + y_{i,t'} \right) \leq u_{t,c}^{Total} \quad \forall c \in C, t \in [0, |T| - c] \tag{4.14}$$

$$\sum_{i \in F} x_{i,t} \beta_{i,s}^{Arr} - \sum_{i \in F} y_{i,t} \beta_{i,s}^{Dep} + n_{t-1,s} = n_{t,s} \quad \forall s \in S, t \in T / \{0\} \tag{4.15}$$

$$\sum_{i \in F} x_{i,0} \beta_{i,s}^{Arr} - \sum_{i \in F} y_{i,0} \beta_{i,s}^{Dep} = n_{0,s} \quad \forall s \in S \tag{4.16}$$

$$n_{t,s} \leq \delta_{t,s} \quad \forall t \in T, s \in S \tag{4.17}$$

$$\sum_{i \in F} \sum_{t' \in [t, t+c-1]} x_{i,t'} e_i \pi_i^{Arr} \leq E_{t,c}^{Arr} \quad \forall t \in [0, |T| - c] \tag{4.18}$$

$$\sum_{i \in F} \sum_{t' \in [t, t+c-1]} y_{i,t'} e_i \pi_i^{Dep} \leq E_{t,c}^{Dep} \quad \forall t \in [0, |T| - c] \tag{4.19}$$

$$\sum_{t' \in T} x_{i,t'} t' + f_i = a_i \quad \forall i \in F \tag{4.20}$$

$$\sum_{t' \in T} y_{i,t'} t' = d_i \quad \forall i \in F \tag{4.21}$$

$$\sum_{g \in G} z_{i,g} = \mu_i \quad \forall i \in F \tag{4.22}$$

$$z_{i,g} \le o_{i,g} \quad \forall i \in F, g \in G \tag{4.23}$$

$$p_i^b \ge a_i \quad \forall i \in F \tag{4.24}$$

$$p_i^e \ge d_i \quad \forall i \in F \tag{4.25}$$

$$p_i^e \ge p_i^b + T_{\min,i} \quad \forall i \in F \tag{4.26}$$

$$p_i^e + \eta - p_i^b \le M\left(2 - z_{i,g} - z_{j,g}\right) \quad \forall i, j \in F,\, g \in G \backslash \{0\}\,,\, i < j \tag{4.27}$$

$$\mu_i \in \{0,1\} \quad \forall i \in F \tag{4.28}$$

$$x_{i,t}, y_{i,t} \in \{0,1\} \quad \forall i \in F, t \in T \tag{4.29}$$

$$n_{t,s} \in Z^+ \cup \{0\} \quad \forall t \in T, s \in S \tag{4.30}$$

$$z_{i,g} \in \{0,1\} \quad \forall i \in F, g \in G \tag{4.31}$$

$$a_i, d_i, p_i^b, p_i^e \ge 0 \quad \forall i \in F \tag{4.32}$$

For the integrated model, the constraints (4.6)-(4.7) ensure that each aircraft's departure (arrival) slot request is either allocated to at most one slot and one terminal, or rejected. Constraints (4.8) ensure that the acceptance or rejection of an aircraft's arrival slot matches its departure slot status. Constraints (4.9) enforce minimum and maximum turnaround times unless the aircraft is rejected. Constraints (4.10) and (4.11) ensure maximum displacement of each aircraft arrival and departure. Runway capacity constraints (4.12)-(4.14) restrict aircraft movements to not exceed runway capacity at any time interval. For apron capacity, the number of parked aircraft ($n_{t,s}$) is calculated using the parameter $\beta_{i,s}^{Arr(Dep)}$. Constraints (4.15)-(4.16) compute the parked-aircraft count at each time step by adding the aircraft already parked to those arriving or departing in that interval. Constraint (4.17) ensure that the apron capacity for each service type is not exceeded. Passenger terminal capacity constraints (4.18)-(4.19) use the weighting factor $\pi_i^{Arr(Dep)}$ to estimate passenger numbers, ensuring these do not exceed terminal capacity. The arrival and departure times of aircraft i

in scenario ω at time interval t are defined in constraints (4.20)-(4.21). Once an aircraft i is accepted ($\mu_i = 1$), it will be assigned to at most one gate, as ensured by constraint (4.22). Compatibility between aircraft and gates is maintained through constraint (4.23). Constraint (4.24) ensures that the parking start time, $p^b_{i,t}$, is not earlier than the arrival time, a^t_i. Similarly, constraint (4.25) guarantees that the parking end time, $p^e_{i,t}$, is not earlier than the departure time, d^t_i. Constraint (4.26) ensures a minimum turnaround time for each aircraft at the gate. To avoid overlaps, Constraint (4.27) requires a buffer of at least η between back-to-back aircraft assigned to the same gate, excluding the temporary apron Gate 0.

4.2.3 Two-stage robust optimization with operational uncertainty

In practice, aircraft arrival times are volatile, and these fluctuations most directly disrupt gate assignment. To address this, we couple slot scheduling with gate assignment under delay uncertainty using a two-stage robust optimization model that safeguards against worst-case operational outcomes. The first stage fixes the flight schedule (slot allocation), while the second stage adapts gate assignments once delays materialize.

Within the robust optimization setting, let f_i denote the realized arrival delay for flight i. We assume $\left[\bar{f}_i - \hat{f}_i, \bar{f}_i + \hat{f}_i\right]$ where $\bar{f}_i$ is the nominal delay and $\hat{f}_i$ is the maximum deviation from nominal. These parameters can be estimated from historical flight operations data, for example by using a central tendency for $\bar{f}_i$ and a high-quantile or envelope for $\hat{f}_i$.

We capture delay uncertainty using the following budget of uncertainty set:

$$\mathrm{K} = \left\{ f | f_i = \bar{f}_i + \xi_i \hat{f}_i, i \in F, \xi \in \Xi \right\} \tag{4.33}$$

where

$$\Xi = \left\{ \xi | \sum_{i \in F} \xi_i \leq \Gamma, 0 \leq \xi_i \leq 1, i \in F, \xi \in \Xi \right\} \tag{4.34}$$

The parameter Γ serves as the budget of uncertainty and tunes the conservatism of the two-stage model. Typically, $\Gamma \in [0, |F|]$. When $\Gamma = 0$, the worst-case realization coincides with all nominal delays, which may be optimistic. When $\Gamma = |F|$, all flights are allowed to hit their maximum deviations simultaneously, yielding a highly conservative plan. Intermediate values $0 \leq \Gamma \leq |F|$ restrict the aggregate deviation so that, in effect, only a limited number of flights can realize their worst-case perturbations. For any chosen Γ, the inner maximization over $\xi \in \Xi$ identifies the worst-case delay vector within the budget, and the outer minimization selects decisions that hedge against that case.

The two-stage robust optimization framework comprises: (i) a first stage that performs airport slot allocation to generate an initial schedule; and (ii) a second stage that assigns gates to that schedule under uncertain flight delays. Specifically:

$$\min_{\mu,x,y} C_{Sr} + C_{Td} + Q(\nu) \tag{4.35}$$

$$\text{s.t. (4.6)-(4.19), (4.28)-(4.30)}$$

where $Q(\nu)$ denotes the optimal value of the second-stage problem. The vector $\nu = \{\mu_i, x_{i,t}, y_{i,t}\}$ collects the first-stage decisions: μ_i specifies the scheduled slot (e.g., planned arrival/departure time) for aircraft i, while $x_{i,t}$ and $y_{i,t}$ indicate the arrival and departure time slots at time interval t, respectively. These decisions are fixed before uncertainty is realized and subsequently inform the second-stage gate assignment and timing adjustments captured by $Q(\nu)$:

$$Q(\nu) = \max_{\xi \in \Xi} \min_{z,a,d,p^b,p^e} C_{Gd} + C_{Ta} \tag{4.36}$$

$$\text{s.t. (4.21)-(4.27), (4.31)-(4.32)}$$

$$\sum_{t' \in T} x_{i,t'} t' + \bar{f}_i + \xi_i \hat{f}_i = a_i \quad \forall i \in F \tag{4.37}$$

In the above second-stage problem, the objective function (4.36) aims to minimize the worst-case total cost, which includes both operational ground delay and the cost of aircraft assigned to the temporary apron. The worst-case scenario is determined by the outer maximization operator, while the inner minimization operator focuses on reducing the operational ground delay and the costs associated with aircraft on the temporary apron.

4.3 Solution approach

For the joint problem, we solve the two-stage robust model with a nested C&CG algorithom that iterates between a master problem and adversarial recourse subproblems. The master problem decides whether to accept or reject aircraft and fixes the planned arrival/departure times (slot allocation). Given these first-stage decisions, the recourse problems, parameterized by worst-case delay realizations, assign gates and adjust gate-occupancy times. The master problem is augmented with cuts and columns from the recourse problems, and the cycle continues to convergence.

The recourse stage has a max–min structure, so standard LP solvers cannot handle it directly. We therefore construct a MIP-based procedure to derive optimal recourse decisions. Structurally, the overall formulation is tri-level with a min–max–min layout. A common reduction—collapsing the inner max–min to a single-level max via duality—requires convexity of the inner minimization, which fails here because it contains binary variables. To overcome this, we adopt a nested C&CG algorithm (see, e.g., Zhao & Zeng (2012)) within a two-tier master–subproblem architecture. The outer C&CG loop addresses the first-stage decisions and worst-case adversarial responses, while the inner (nested) C&CG resolves the second-stage max–min recourse with integrality, yielding valid cuts and ensuring convergence.

As mentioned in Section 4.2.3, the budgeted uncertainty set K generates a family of feasible delay realizations controlled by Γ. Each realization is represented by a vector $\xi \in \Xi$, and for computational purposes we treat these realizations as a finite collection of scenarios indexed by Ω. For any scenario $\omega \in \Omega$, we write $\xi(\omega)$ for the associated deviation pattern. Given a fixed first-stage plan, scenario ω induces a recourse adjustment characterized by $(a_i(\omega), d_i(\omega), p_i^b(\omega), p_i^e(\omega))$ for each aircraft i. Here, $a_i(\omega)$ and $d_i(\omega)$ capture the realized gate-occupancy start and end times (or adjusted gate arrival and departure), and $p_i^b(\omega)$ and $p_i^e(\omega)$ encode the discrete gate-assignment choices and conflict-resolution indicators that ensure feasibility under the delay vector implied by $f_i = \bar{f}_i + \xi_i(\omega)\hat{f}_i$.

We formulate the problem as a MIP to compute the optimal solution. The first-stage variables (e.g., $\mu_i, x_{i,t}, y_{i,t}$) fix aircraft acceptance and nominal slot times before uncertainty is realized; these decisions are shared by all scenarios. For each scenario, the recourse submodel adjusts gate occupancy and assignments using $(a_i(\omega), d_i(\omega), p_i^b(\omega), p_i^e(\omega))$, honoring gate capacity and turnaround constraints that are parameterized by the realized delays. The overall objective evaluates the induced costs across scenarios and selects first-stage decisions that hedge against adverse deviations, typically by minimizing the maximum cost over Ω. Within a C&CG framework, only the most critical scenarios—those exposing the largest violations or costs—are iteratively added, and the process continues until no scenario can further worsen the objective, yielding a robust optimal solution.

$$\min_{\mu,x,y} C_{Sr} + C_{Td} + \theta \tag{4.38}$$

s.t. (4.6)-(4.19), (4.28)-(4.30)

$$\theta \geq \sum_{i \in F} p_i^b(\omega) - a_i(\omega) + p_i^e(\omega) - d_i(\omega) + \sigma z_{i,0}(\omega) \quad \forall \omega \in \Omega \tag{4.39}$$

$$\sum_{t' \in T} x_{i,t'} t' + \bar{f}_i + \xi_i(\omega) \hat{f}_i = a_i(\omega) \quad \forall i \in F, \omega \in \Omega \tag{4.40}$$

$$\sum_{t' \in T} y_{i,t'} t' = d_i(\omega) \quad \forall i \in F, \omega \in \Omega \tag{4.41}$$

$$\sum_{g \in G} z_{i,g}(\omega) = \mu_i \quad \forall i \in F, \omega \in \Omega \tag{4.42}$$

$$z_{i,g}(\omega) \leq o_{i,g} \quad \forall i \in F, g \in G, \omega \in \Omega \tag{4.43}$$

$$p_i^b(\omega) \geq a_i(\omega) \quad \forall i \in F, \omega \in \Omega \tag{4.44}$$

$$p_i^e(\omega) \geq d_i(\omega) \quad \forall i \in F, \omega \in \Omega \tag{4.45}$$

$$p_i^e(\omega) \geq p_i^b(\omega) + T_{\min,i} \quad \forall i \in F, \omega \in \Omega \tag{4.46}$$

$$p_i^e(\omega) + \eta - p_i^b(\omega) \leq M\left(2 - z_{i,g}(\omega) - z_{j,g}(\omega)\right) \quad \forall i, j \in F,\, g \in G \setminus \{0\},\, i < j, \omega \in \Omega \tag{4.47}$$

$$z_{i,g}(\omega) \in \{0,1\} \quad \forall i \in F, g \in G, \omega \in \Omega \tag{4.48}$$

$$a_i(\omega), d_i(\omega), p_i^b(\omega), p_i^e(\omega) \geq 0 \quad \forall i \in F, \omega \in \Omega \tag{4.49}$$

The number of scenarios Ω for the above problem grows exponentially, making it unsolvable directly using a solver. The C&CG algorithm addresses this by solving the simplified MIP model, known as the master problem (MP), iteratively. In this process, constraints (4.39)-(4.49) are replaced by a finite subset $\omega^k \in \Omega$, $k = 1, \ldots, K$. The MP is shown as follows:

$$\textbf{MP:} \quad \min_{\mu,x,y,\theta,z_{i,g},a_i,d_i,p_i^b,p_i^e} C_{Sr} + C_{Td} + \theta \tag{4.50}$$

$$\text{s.t. (4.6)-(4.19), (4.28)-(4.30)}$$

$$\theta \geq \sum_{i \in F} p_i^{b,k} - a_i^k + p_i^{e,k} - d_i^k + \sigma z_{i,0}^k \quad \forall k = 1, \ldots, K \tag{4.51}$$

$$\sum_{t' \in T} x_{i,t'} t' + \bar{f}_i + \xi_i^k \hat{f}_i = a_i^k \quad \forall i \in F, k = 1, \ldots, K \tag{4.52}$$

$$\sum_{t' \in T} y_{i,t'} t' = d_i^k \quad \forall i \in F, k = 1, \ldots, K \tag{4.53}$$

$$\sum_{g \in G} z_{i,g}^k = \mu_i \quad \forall i \in F, k = 1, \ldots, K \tag{4.54}$$

$$z_{i,g}^k \leq o_{i,g} \quad \forall i \in F, g \in G, k = 1, \ldots, K \tag{4.55}$$

$$p_i^{b,k} \geq a_i^k \quad \forall i \in F, k = 1, \ldots, K \tag{4.56}$$

$$p_i^{e,k} \geq d_i^k \quad \forall i \in F, k = 1, \ldots, K \tag{4.57}$$

$$p_i^{e,k} \geq p_i^{b,k} + T_{\min,i} \quad \forall i \in F, k = 1, \ldots, K \tag{4.58}$$

$$p_i^{e,k} + \eta - p_i^{b,k} \leq M\left(2 - z_{i,g}^k - z_{j,g}^k\right) \quad \forall i, j \in F,\, g \in G \backslash \{0\}, i < j, k = 1, \ldots, K \tag{4.59}$$

$$z_{i,g}^k \in \{0,1\} \quad \forall i \in F, g \in G, k = 1, \ldots, K \tag{4.60}$$

$$a_i^k, d_i^k, p_i^{b,k}, p_i^{e,k} \geq 0 \quad \forall i \in F, k = 1, \ldots, K \tag{4.61}$$

In each iteration, the solution value $C_{Sr} + C_{Td} + \theta$ obtained by solving the MP is used as the lower bound of the optimal objective value. The recourse problem SP(ν) is then solved using the current first-stage solution $\nu = \{\mu_i, x_{i,t}, y_{i,t}\}$. The SP($\nu$) is shown as follows:

$$\textbf{SP}(\nu)\textbf{:} \quad \max_{\xi \in \Xi} \min_{z,a,d,p^b,p^e} C_{Gd} + C_{Ta} \tag{4.62}$$

s.t. (4.21)-(4.27), (4.31)-(4.32), (4.37)

After solving the recourse subproblem, we either identify a worst-case realization ω^* with associated recourse decisions $\left(z_{i,g}, a_i, d_i, p_i^b, p_i^e\right)$, or we uncover a realization ω^* that renders the subproblem infeasible under the current first-stage plan. In the first case, combining the first-stage objective with the recourse cost at ω^* provides an incumbent upper bound on the robust objective; in the second, the infeasibility certificate reveals a violating scenario requiring attention.

In both situations, ω^* is used to enrich the current master problem by augmenting the active scenario set ω^k and appending the corresponding variables, linking constraints, and valid inequalities implied by the recourse model under ω^*. The master problem is then re-optimized to update the lower bound through an improved first-stage plan and an updated underestimator of the worst-case response. We alternate between solving the master and recourse problems until the relative UB–LB gap is less than ε, after which the algorithm terminates with a robust optimal solution.

Algorithm 3 The outer-level C&CG Algorithm

Input: coordination parameters u, δ, E, sets F, T, C, S
Output: MP solution v^*, SP solutions z^*
1: **Initilize** low bound $LB = -\infty$, upper bound $UB = \infty$, k = 0, $\omega^k \leftarrow \emptyset$, $\omega^* \leftarrow None$
2: **while** $\frac{UB-LB}{LB} \geq \varepsilon$ **do**
3: Extend the **MP** with scenario ω^*
4: Solve the **MP** to obtain the optimal solution $(v^{(k+1)^*}, \theta^{(k+1)^*}, z^{1^*}, \ldots, z^{k^*})$
5: Update $LB \leftarrow max\left\{LB, C_{Sr} + C_{Td} + \theta^{(k+1)^*}\right\}$
6: Solve SP($v^{(k+1)^*}$) to obtain the objective value $Q(v^{(k+1)^*})$
7: Update $UB \leftarrow min\left\{UB, C_{Sr} + C_{Td} + Q(v^{(k+1)^*})\right\}$
8: Update the current worse-case scenario ω^* and $k = k+1$
9: **end while**

The outer-level C&CG converts the two-stage RO into a monolithic MILP over a growing scenario set, but the subproblem remains hard because integrality breaks strong duality, making standard bi-level reductions inapplicable. We therefore recast the bi-level subproblem as a tri-level program that separates integer and continuous decisions. Using an inner-level C&CG, we reformulate the tri-level structure into an equivalent single-level model. By isolating the integer variables, the subproblem becomes a min–max–min problem, which the inner loop solves to optimality by iteratively adding cuts and variables.

$$\max_{\xi\in\Xi}\min_{z}\sum_{i\in F}\sigma z_{i,0}+\min_{a,d,p^b,p^e}\sum_{i\in F}p_i^b-a_i+p_i^e-d_i \tag{4.63}$$
$$\text{s.t. (4.21)-(4.27), (4.32), (4.37)}$$

By exploiting the fact that the integer variable z takes values from a countable set, the above problem can be reformulated equivalently as:

$$\max_{\xi\in\Xi}\pi \tag{4.64}$$
$$s.t.\pi\leq\sum_{i\in F}\sigma z_{i,0}^u+p_i^{b,u}-a_i^u+p_i^{e,u}-d_i^u\quad\forall u=1,\dots,U \tag{4.65}$$
$$\sum_{t'\in T}x_{i,t'}t'+\bar{f}_i+\xi_i^u\hat{f}_i=a_i^u\quad\forall i\in F,u=1,\dots,U \tag{4.66}$$
$$\sum_{t'\in T}y_{i,t'}t'=d_i^u\quad\forall i\in F,u=1,\dots,U \tag{4.67}$$
$$\sum_{g\in G}z_{i,g}^u=\mu_i\quad\forall i\in F,u=1,\dots,U \tag{4.68}$$
$$z_{i,g}^u\leq o_{i,g}\quad\forall i\in F,g\in G,u=1,\dots,U \tag{4.69}$$
$$p_i^{b,u}\geq a_i^u\quad\forall i\in F,u=1,\dots,U \tag{4.70}$$
$$p_i^{e,u}\geq d_i^u\quad\forall i\in F,u=1,\dots,U \tag{4.71}$$
$$p_i^{e,u}\geq p_i^{b,u}+T_{\min,i}\quad\forall i\in F,u=1,\dots,U \tag{4.72}$$
$$p_i^{e,u}+\eta-p_i^{b,u}\leq M\left(2-z_{i,g}^u-z_{j,g}^u\right)\quad\forall i,j\in F,\,g\in G\backslash\{0\},\,i<j,u=1,\dots,U \tag{4.73}$$
$$a_i^u,d_i^u,p_i^{b,u},p_i^{e,u}\geq 0\quad\forall i\in F,u=1,\dots,U \tag{4.74}$$

where $\left\{a_i^u,d_i^u,p_i^{b,u},p_i^{e,u}\right\}$ are the corresponding decision variable for z^u. Let $\lambda_i^u,\phi_i^u,\alpha_i^u,\beta_i^u,\gamma_i^u$ and $\delta_{i,j,g}^u$ represent the dual variables associated with constraints (4.66) (4.67) and (4.70) - (4.73), the problem is equivalently reformulated as:

$$\max_{\xi \in \Xi} \pi \tag{4.75}$$

$$s.t. (4.65) - (4.74)$$

$$-1 + \lambda_i^u - \alpha_i^u \geq 0 \quad \forall i \in F, u = 1, \ldots, U \tag{4.76}$$

$$-1 + \phi_i^u - \beta_i^u \geq 0 \quad \forall i \in F, u = 1, \ldots, U \tag{4.77}$$

$$1 - \alpha_i^u - \gamma_i^u + \sum_{j,g} \delta_{i,j,g}^u \geq 0 \quad \forall i \in F, u = 1, \ldots, U \tag{4.78}$$

$$1 - \beta_i^u - \gamma_i^u - \sum_{j,g} \delta_{i,j,g}^u \geq 0 \quad \forall i \in F, u = 1, \ldots, U \tag{4.79}$$

$$\alpha_i^u (p_i^{b,u} - a_i^u) = 0 \quad \forall i \in F, u = 1, \ldots, U \tag{4.80}$$

$$\beta_i^u (p_i^{e,u} - d_i^u) = 0 \quad \forall i \in F, u = 1, \ldots, U \tag{4.81}$$

$$\gamma_i^u (p_i^{e,u} - p_i^{b,u} - T_{min,i}) = 0 \quad \forall i \in F, u = 1, \ldots, U \tag{4.82}$$

$$\delta_{i,j,g}^u \left[-p_i^{e,u} + p_i^{b,u} - (\eta - M(2 - z_{i,g}^u - z_{j,g}^u)) \right] = 0$$

$$\forall i, j \in F,\ g \in G \setminus \{0\},\ i < j, u = 1, \ldots, U \tag{4.83}$$

$$\lambda_i^u, \phi_i^u, \alpha_i^u, \beta_i^u, \gamma,_i^u \geq 0 \quad \forall i \in F, u = 1, \ldots, U \tag{4.84}$$

$$\delta_{i,j,g}^u \geq 0 \quad \forall i, j \in F, g \in G, u = 1, \ldots, U \tag{4.85}$$

To linearize the constraints (4.80)–(4.83), we introduce binary variables along with the Big-M method. As a result, we derive the final master problem of SP, denoted as SP_m:

$$\max_{\xi \in \Xi} \pi \tag{4.86}$$

s.t.(4.65) - (4.74), (4.76) - (4.79), (4.84) - (4.85)

$$\alpha_i^u \leq M u_i^u \quad \forall i \in F, u = 1, \ldots, U \tag{4.87}$$

$$p_i^{b,u} - a_i^u \leq M(1 - u_i^u) \quad \forall i \in F, u = 1, \ldots, U \tag{4.88}$$

$$\beta_i^u \leq M v_i^u, \quad \forall i \in F, u = 1, \ldots, U \tag{4.89}$$

$$p_i^{e,u} - d_i^u \leq M(1 - v_i^u) \quad \forall i \in F, u = 1, \ldots, U \tag{4.90}$$

$$\gamma_i^u \leq M w_i^u \quad \forall i \in F, u = 1, \ldots, U \tag{4.91}$$

$$p_i^{e,u} - p_i^{b,u} - T_{min,i} \leq M(1 - w_i^u) \quad \forall i \in F, u = 1, \ldots, U \tag{4.92}$$

$$\delta_{i,j,g}^u \leq M q_{i,j,g}^u \quad \forall i, j \in F,\ g \in G \setminus \{0\}, i < j, u = 1, \ldots, U \tag{4.93}$$

$$-p_i^{e,u} + p_i^{b,u} - (\eta - M(2 - z_{i,g}^u - z_{j,g}^u)) \leq M(1 - q_{i,j,g}^u)$$

$$\forall i, j \in F,\ g \in G \setminus \{0\}, i < j, u = 1, \ldots, U \tag{4.94}$$

$$u_i^u, v_i^u, w_i^u, q_{i,j,g}^u \in \{0, 1\} \tag{4.95}$$

Similar to outer-level C&CG algorithm, we anticipate that enumerating all possible variables and constraints for every z^u is unnecessary. Consequently, the inner-level C&CG algorithm is outlined in Algorithm 4, is designed to address this efficiently.

Algorithm 4 The inner-level C&CG Algorithm

Input: MP's solution ν, sets F, G
Output: Optimal solution $\left(\xi^*, z^*, p^{b^*}, p^{e^*}\right)$

Initilize low bound $LB = -\infty$, upper bound $UB = \infty$, u = 0
while $\frac{UB-LB}{LB} \geq \varepsilon$ **do**
  Extend the SP_m by including the constraints of the subproblem of SP_m
  Solve the SP_m to obtain the optimal solution ξ^* and objective value $Q(\nu)$
  Update $UB \leftarrow min\{UB, Q(\nu)\}$
  Solve the subproblem of SP_m to obtain the optimal solution $\left(z^{u+1}, a^{u+1}, d_i^{u+1}, p^{b,u+1}, p^{e,u+1}\right)$
  Update $LB \leftarrow max\left\{LB, C_{Gr}^{u+1} + C_{Ta}^{u+1})\right\}$
  Update $u = u + 1$
end while

4.4 Numerical studies

We evaluate the proposed model and algorithms on real-world instances. We implement all models in Python and solve them using Gurobi 11 (LP/MIP). Experiments are conducted on a 26-core 2.1 GHz Intel Xeon workstation with 128 GB RAM.

4.4.1 Instance generation

All simulation data are drawn from the 2019 summer flight schedule and realized arrival/departure records for Shanghai Hongqiao Airport (SHA). Since real slot request data are not public, we approximate requests with the planned schedule, defining schedule displacement as the absolute gap between the allocated slot and the planned time. In the absence of published slot-priority classes, we assume uniform priority across all requests.

We partition each day into 288 five-minute intervals. The maximum displacement Ψ is 60 minutes, and the minimum turnaround time is 65 minutes. Penalty coefficients are set to ζ = 50 and σ = 288. A curfew is in effect from 00:00 to 06:00. To model uncertainty in flight delays, we fit distributions using kernel density estimation and normal fits to 2019 historical data, and derive aircraft-specific arrival and departure probability distributions for each $i \in F$.

We simulated airport infrastructure and capacity using SHA's capabilities to generate three sets of cases, representing small and medium airports. Table 4.2 shows the declared capacity of each airport. For each set of cases, we simulated idle, normal, and congested airport conditions by setting different aircraft numbers: $F_S = \{25, 50, 100\}$ and $F_L = \{100, 175, 250\}$.

Table 4.2: Parameter settings for different Airports

Parameter (notation)	Description	Small	Large
Runway Capacity ($u_{t,c}^{Arr(Dep)}$)	Arrivals	4 per 60 min	15 per 60 min
	Departures	4 per 60 min	15 per 60 min
	Total	6 per 60 min	22 per 60 min
Passenger Terminal Capacity ($E_{t,c}^{Arr(Dep)}$)	Arrivals	900 per 60 min	3600 per 60 min
	Departures	900 per 60 min	3600 per 60 min
Apron Capacity ($\delta_{t,s}$)	Passenger	14	49
	Cargo	1	1
Turnaround Time ($T_{\min,i}/T_{\max,i}$)	Minimum/Maximum	65 min / 180 min	65 min / 180 min
Maximum Displacement (Ψ)	-	60 min	60 min
Load Factor ($\pi_i^{Arr(Dep)}$)	-	0.8	0.8
Gates (g)	Passenger	14	49
	Cargo	1	1
Parking Interval (η)	Minimum	20 min	20 min

4.4.2 Computational results

Table 4.3 provides the calculation process for a small airport under varying numbers of aircraft and levels of budget uncertainty, denoted as Γ. The results highlight the impact of these two factors on airport operations, computational time, and associated costs. With more aircraft, airport congestion intensifies, stretching limited resources and substantially increasing optimization runtimes. This increase in congestion leads to higher operational costs due to the added complexity of managing a greater number of flights within constrained resources.

Similarly, as budget uncertainty Γ increases, the complexity of managing airport operations grows substantially. Higher levels of Γ introduce greater variability and unpredictability in resource allocation and scheduling, necessitating more sophisticated process. This results in a notable rise in both computation time and costs, as the model must account for a wider range of scenarios and potential disruptions.

Table 4.3: Computational results for small airport varying Γ

F	Γ	Obj	Out-level iteration	Inner-level iteration	Time(s)
25	5	376	3	4,3,3	32
	15	376	3	2,5,3	33
	25	379	3	2,3,4	31
50	5	537	4	3,4,5,3	38
	25	542	5	6,7,5,3,5	49
	50	550	7	3,9,4,4,5,6,2	55
100	25	731	12	34,54,12,25,21,11, 20,26,12,14,33,32	320
	50	792	15	46,21,10,12,31,41,20, 12,16,18,17,21,32,10	396
	100	861	16	32,22,15,4,34,12,6,14, 48,16,35,32,19,21,22,10	423

Table 4.4 presents the computational process for a large airport under varying aircraft numbers and levels of budget uncertainty. The results align closely with those observed for the small airport: as the number of aircraft increases, congestion at the airport intensifies, leading to longer computation times. Similarly, higher levels of budget uncertainty add complexity to the operational environment, resulting in increased total costs and longer computation times.

Table 4.4: Computational results for large airport varying Γ

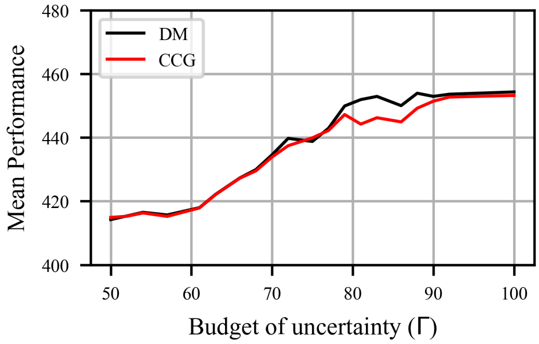
F	Γ	Obj	Out-level iteration	Gap	Time(s)
100	50	437	2	0	262
	60	435	3	0.00%	260
	70	435	2	0.00%	259
	80	443	3	0.01%	266
	90	448	3	0.01%	259
	100	445	3	0.01%	273
175	50	794	10	0.01	367
	75	819	12	0.02%	375
	100	827	13	0.10%	365
	125	827	11	0.24%	397
	150	829	14	0.09%	375
	175	836	16	0.06%	370
250	50	1418	34	0.88	2751
	80	1431	39	0.69%	2988
	110	1440	35	0.47%	2896
	140	1452	34	0.63%	3021
	170	1463	31	0.55%	2999
	200	1462	42	0.26%	2956
	230	1487	46	0.67%	2874
	250	1486	46	0.78%	2765

4.4.3 Performance with deterministic model and the robust optimization model

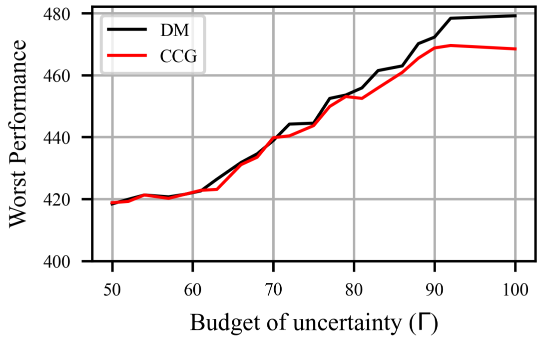
We generate $|\Omega| = 500$ random scenarios to assess the proposed two-stage RO model. Using the instances $F_L = 100$ and $F_L = 250$ from Section 4.4.1, we compare mean and worst-case performance of the deterministic and robust optimization models under these scenarios. For $F_L = 100$, Γ varies from 50 to 100; for $F_L = 250$, Γ varies from 50 to 250.

Fig. 4.1 reports the outcomes. When uncertainty is mild ($\Gamma < 100$), the deterministic model achieves lower mean and worst-case cost. For $100 \leq \Gamma < 150$, performances are similar. Once $\Gamma \geq 150$, the two-stage RO model yields lower mean and worst-case cost.

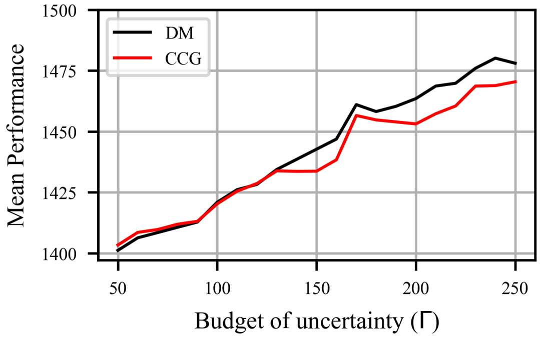
Overall, the figure shows no uniform dominance at small budgets, whereas at larger budgets the RO model consistently performs better. This pattern stems from discrete first-stage choices: the two models diverge only when uncertainty crosses certain breakpoints. As Γ increases—indicating greater volatility—the hedging value of RO becomes more pronounced.



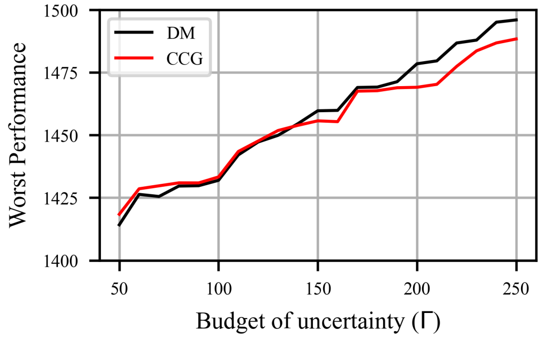
(a) Mean Performance $F_L = 100$



(b) Worst Performance $F_L = 100$



(c) Mean Performance $F_L = 250$



(d) Worst Performance $F_L = 250$

Figure 4.1: Performances of deterministic and two-stage RO models with different budget of uncertainty Γ

4.4.4 Sensitivity analysis

This section presents the parameter sensitivity analysis. Using the instance with $F_L = 250$, we execute the C&CG algorithm under controlled perturbations of key inputs and report the cost components defined in Section 4.2.2. This makes it possible to quantify how parameter shifts affect total, first-stage, and recourse costs and to determine which parameters have the strongest marginal effect on solution quality.

The first parameter is ζ, the penalty for rejecting a slot request. We assess its effect on the four cost components $(C_{Sr}, C_{Td}, C_{Gd}, C_{Ta})$ by varying ζ over [40, 60]. As shown in Figure 4.2, all cost categories rise overall except C_{Sr}. A higher rejection penalty motivates the operator to accept more requests, which increases slot displacement (C_{Td}). Greater acceptance also assigns more aircraft to gates, intensifying operational congestion and elevating gate-related costs ($C_{Gd} + C_{Ta}$). Notably, the total cost ($C_{Sr} + C_{Td} + C_{Gd} + C_{Ta}$) remains unchanged, reflecting efforts to minimize the overall objective.

(a) C_{Sr}: the cost of rejecting slot requests

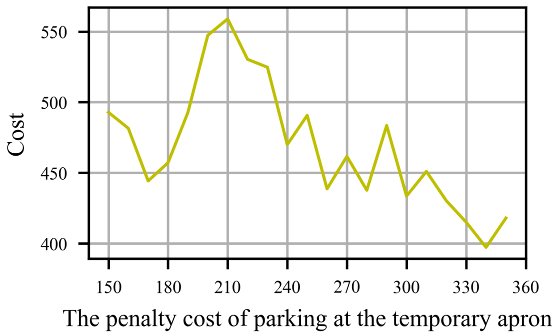
(b) C_{Td}: the cost of total displacement

(c) $C_{Gd} + C_{Ta}$: the total cost of operation

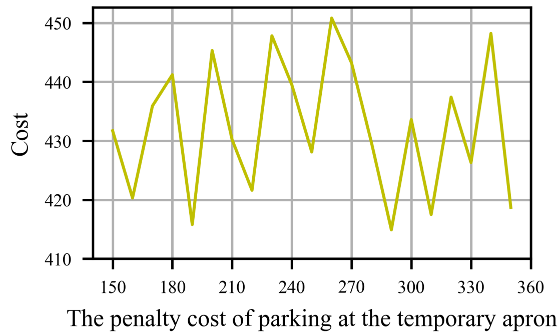
(d) $C_{Sr} + C_{Td} + C_{Gd} + C_{Ta}$: total cost

Figure 4.2: The results for four cost types under different parameter ζ

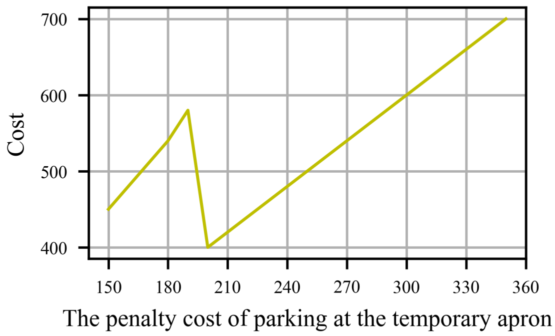
Another important parameter is σ, the penalty cost of parking at the temporary apron (Gate 0). We examine its impact on the four types of cost ($C_{Sr}, C_{Td}, C_{Gd}, C_{Ta}$) by adjusting the penalty cost from 150 to 350. Figure 4.3 indicates that as the penalty cost of parking on a temporary apron increases, both the total planning stage cost and the ground delay cost of the operations stage rise ($C_{Sr} + C_{Td}, C_{Gd}$), while the cost of parking on a temporary apron decreases (C_{Ta}). This suggests that higher penalty costs prompt airports to more actively reject slot requests during the planning stage to achieve a less congested operating environment. Airports will also be more proactive in delaying aircraft movements into or out of gates to avoid using the temporary apron.



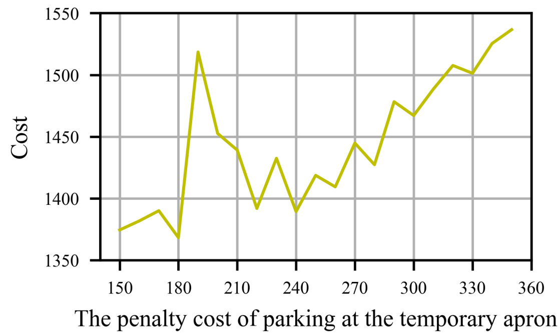
(a) $C_{Sr} + C_{Td}$: the cost of planning



(b) C_{Td}: the cost of ground delay



(c) C_{Ta}: temporary apron parking cost



(d) $C_{Sr} + C_{Td} + C_{Gd} + C_{Ta}$: total cost

Figure 4.3: The results for four cost types under different parameter σ

4.5 Summary

This study proposes a unified modeling framework that jointly optimizes planning-level scheduling (ASA) and post-realization gate assignment at the operational level. The problem is cast as a two-stage robust optimization model designed to hedge against worst-case scenarios and guarantee feasibility under severe disruptions. By incorporating robustness into both stages of the optimization process, the framework offers a practical and resilient approach to airport scheduling. To efficiently handle the computational burden of large-scale, worst-case scenarios, the algorithm employs the nested C&CG algorithm.

We validated the proposed model with real-world data from SHA. The results demonstrated the algorithm's scalability and effectiveness, delivering high-quality robust solutions within practical computation times. The Convergence results for the nested C&CG algorithm highlight its efficiency in tackling complex, large-scale instances, despite significant uncertainty. The case study shows that the robust optimization framework offers actionable decision support for coordinated scheduling, bolstering resilience, alleviating congestion, and improving reliability. Furthermore, the framework effectively captures the critical interdependencies between strategic planning and operational adaptation, offering a holistic solution to the challenges of airport management under uncertainty.

This research opens several promising directions for future exploration. First, while this study focuses on slot allocation and gate assignment for a single airport, extending the framework to a network of interconnected airports could offer deeper insights into system-wide scheduling and allocation under uncertainty. Second, while the robust optimization framework prioritizes resilience to worst-case scenarios, future research could explore hybrid approaches that combine robust and stochastic optimization to achieve a balance between worst-case resilience and probabilistic adaptability. Finally, further advancements in algorithmic efficiency, such as integrating machine learning techniques, could enable the application of this methodology to even larger and more complex instances, supporting robust and scalable decision-making for airports of all sizes.

Chapter 5

Conclusion

5.1 Major finding

The thesis addresses the critical challenges of integrating ASA with short-term gate assignment under uncertainty and proposes two innovative modeling frameworks to tackle these issues. The first framework is a 2SSIP model designed to handle probabilistic uncertainties such as stochastic flight delays. In this model, slot allocation is optimized in the first-stage, while gate assignments are dynamically adjusted in the second-stage using the rolling horizon framework. Incorporating dual offline cuts from the second-stage LP relaxation tightens global lower bounds and improves computational efficiency. The addition of neighborhood constraints further hastens convergence of the decomposition algorithm. With Shanghai Hongqiao Airport data, the 2SSIP model managed instances as large as 20 million variables and 6 billion constraints in practical runtimes. The findings verify the model's responsiveness to real-time disruptions and its skill in balancing long-term planning with operational adjustments.

The second framework adopts a two-stage robust optimization model that safeguards operations against worst-case conditions, including severe weather and ripple-effect delays. This framework incorporates robustness into both stages of the optimization process, offering a highly resilient approach to airport scheduling. A nested C&CG algorithm is employed to efficiently fix large-scale robust optimization problems, iteratively refining the solution space to achieve faster convergence. The algorithm demonstrated scalability and effectiveness in solving complex instances while maintaining resilience against worst-case disruptions. The case study at Shanghai Hongqiao Airport underscores the robustness and practicality of the proposed model in mitigating congestion, ensuring operational reliability, and enhancing airport resilience.

Together, these two complementary frameworks provide a comprehensive approach for addressing uncertainty in airport scheduling. The stochastic model excels in scenarios requiring probabilistic adaptability, while the robust model ensures resilience to extreme disruptions. Both frameworks emphasize the critical interdependencies between long-term strategic planning and real-time operational decision-making, offering actionable solutions for congested airports of various sizes.

5.2 Future work

Although this thesis advances the state of airport scheduling, it also points to promising future work. An important avenue is to generalize the frameworks to network-wide scheduling and coordination. This would involve optimizing slot allocation and gate assignment across multiple interconnected airports, providing a broader perspective on system-wide scheduling under uncertainty. Such extensions could address cascading disruptions across the network, contributing to more efficient and resilient air transportation systems.

Advancements in algorithmic efficiency also represent a key area for future research. Incorporating machine learning techniques, such as predictive modeling for uncertainty estimation, could enhance the performance of the proposed models. Additionally, parallel computing and distributed optimization techniques could further reduce computation times, enabling the application of these frameworks to even larger and more complex instances.

Future work could also consider integrating additional operational constraints, such as passenger flow optimization, taxiway congestion, and environmental factors like noise and emissions limits. Expanding the models to capture these complexities would provide a more comprehensive view of airport operations and enhance their practical applicability. Finally, implementing these frameworks in real-time operational environments would offer valuable insights into their practical utility and effectiveness. Collaborations with airport operators could refine the models and ensure alignment with the needs of modern airport management.

By tackling these avenues, future work can capitalize on the thesis's insights to push the frontier of airport scheduling, helping develop air transportation systems that are more efficient, reliable, and resilient amid increasing congestion and operational volatility.

References

ACI World Store (2024). 2024 Annual World Airport Traffic Dataset. `https://store.aci.aero/product/annual-world-airport-traffic-dataset-2024/`.

Andreatta, G., & Romanin-Jacur, G. (1987). Aircraft flow management under congestion. *Transportation Science*, *21*, 249–253.

Aoun, O., & El Afia, A. (2014). Using Markov decision processes to solve stochastic gate assignment problem. In *2014 International Conference on Logistics Operations Management* (pp. 42–47).

Benlic, U. (2018). Heuristic search for allocation of slots at network level. *Transportation Research Part C: Emerging Technologies*, *86*, 488–509.

Benlic, U., Burke, E. K., & Woodward, J. R. (2017). Breakout local search for the multi-objective gate allocation problem. *Computers & Operations Research*, *78*, 80–93.

Bi, J., Wang, F., Ding, C., Xie, D., & Zhao, X. (2022). The airport gate assignment problem: A branch-and-price approach for improving utilization of jetways. *Computers & Industrial Engineering*, *164*, 107878.

Birge, J. R., & Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Birolini, S., Jacquillat, A., Schmedeman, P., & Ribeiro, N. (2023). Passenger-centric slot allocation at schedule-coordinated airports. *Transportation Science*, *57*, 4–26.

Bouras, A., Ghaleb, M. A., Suryahatmaja, U. S., & Salem, A. M. (2014). The Airport Gate Assignment Problem: A Survey. *The Scientific World Journal*, *2014*, e923859.

Castelli, L., Pellegrini, P., & Pesenti, R. (2012). Airport slot allocation in europe: economic efficiency and fairness. *International journal of revenue management*, *6*, 28–44.

Chandran, B., & Balakrishnan, H. (2007). A Dynamic Programming Algorithm for Robust Runway Scheduling. In *2007 American Control Conference* (pp. 1161–1166). ISSN: 2378-5861.

Cheng, C.-H., Ho, S. C., & Kwan, C.-L. (2012). The use of meta-heuristics for airport gate assignment. *Expert systems with applications*, *39*, 12430–12437.

Civil Aviation Administration of China (CAAC) (2024). Statistics on major transportation production indicators of china civil aviation in 2023. `https://app.caac.gov.cn/PHONE/XXGK_17/XXGK/TJSJ/202405/P020240531700964611105.pdf`.

Corolli, L., Lulli, G., & Ntaimo, L. (2014). The time slot allocation problem under uncertain capacity. *Transportation Research Part C: Emerging Technologies*, *46*, 16–29.

Daş, G. S., Gzara, F., & Stützle, T. (2020). A review on airport gate assignment problems: Single versus multi objective approaches. *Omega*, *92*, 102146.

Du, W., Zhu, S., Tong, L., Cai, K., & Liang, Z. (2023). Robust gate assignment to minimise aircraft conflicts. *Transportmetrica B: Transport Dynamics*, *11*, 2185497.

Evler, J., Asadi, E., Preis, H., & Fricke, H. (2021). Airline ground operations: Schedule recovery optimization approach with constrained resources. *Transportation Research Part C: Emerging Technologies*, *128*, 103129.

Fairbrother, J., & Zografos, K. G. (2021). Optimal scheduling of slots with season segmentation. *European Journal of Operational Research*, *291*, 961–982.

Fairbrother, J., Zografos, K. G., & Glazebrook, K. D. (2020). A Slot-Scheduling Mechanism at Congested Airports that Incorporates Efficiency, Fairness, and Airline Preferences. *Transportation Science*, *54*, 115–138.

Fan, T. P., & Odoni, A. R. (2002). A practical perspective on airport demand management. *Air Traffic Control Quarterly*, *10*, 285–306.

Glomb, L., Liers, F., & Rösel, F. (2022). A rolling-horizon approach for multi-period optimization. *European Journal of Operational Research*, *300*, 189–206.

Glover, F., & Laguna, M. (1998). *Tabu search*. Springer.

Hassounah, M. I., & Steuart, G. N. (1993). Demand for aircraft gates. *Transportation research record*, .

International Air Transport Association (IATA) (2024). Worldwide Airport Slot Guidelines (WASG). `https://www.iata.org/en/programs/ops-infra/slots/slot-guidelines`.

Jiang, Y., Hu, Z., Liu, Z., & Zhang, H. (2023). Optimization of multi-objective airport gate assignment problem: considering fairness between airlines. *Transportmetrica B: Transport Dynamics*, *11*, 196–210.

Jiang, Y., & Zografos, K. G. (2021). A decision making framework for incorporating fairness in allocating slots at capacity-constrained airports. *Transportation Research Part C: Emerging Technologies*, *126*, 103039.

Jiefeng, X., & Bailey, G. (2001). The airport gate assignment problem: mathematical model and a tabu search algorithm. *System Sciences*, .

Jorge, D., Ribeiro, N. A., & Antunes, A. P. (2021). Towards a decision-support tool for airport slot allocation: Application to guarulhos (sao paulo, brazil). *Journal of Air Transport Management*, *93*, 102048.

Karsu, Ö., Azizoğlu, M., & Alanlı, K. (2021). Exact and heuristic solution approaches for the airport gate assignment problem. *Omega*, *103*, 102422.

Karsu, Ö., & Solyalı, O. (2023). A new formulation and an effective matheuristic for the airport gate assignment problem. *Computers & Operations Research*, *151*, 106073.

Katsigiannis, F. A., & Zografos, K. G. (2021). Optimising airport slot allocation considering flight-scheduling flexibility and total airport capacity constraints. *Transportation Research Part B: Methodological*, *146*, 50–87.

Katsigiannis, F. A., & Zografos, K. G. (2023). Incorporating slot valuation in making airport slot scheduling decisions. *European Journal of Operational Research*, *308*, 436–454.

Keskin, M., & Zografos, K. G. (2023). Optimal network-wide adjustments of initial airport slot allocations with connectivity and fairness objectives. *Transportation Research Part B: Methodological*, *178*, 102801.

Khassiba, A., Bastin, F., Cafieri, S., Gendron, B., & Mongeau, M. (2020). Two-stage stochastic mixed-integer programming with chance constraints for extended aircraft arrival management. *Transportation Science*, *54*, 897–919.

Khassiba, A., Cafieri, S., Bastin, F., Mongeau, M., & Gendron, B. (2022). Two-stage stochastic programming models for the extended aircraft arrival management problem with multiple pre-scheduling points. *Transportation Research Part C: Emerging Technologies*, *142*, 103769.

Kim, J., Goo, B., Roh, Y., Lee, C., & Lee, K. (2023). A branch-and-price approach for airport gate assignment problem with chance constraints. *Transportation Research Part B: Methodological*, *168*, 1–26.

Kim, S. H., Feron, E., Clarke, J.-P., Marzuoli, A., & Delahaye, D. (2017). Airport gate scheduling for passengers, aircraft, and operations. *Journal of Air Transportation*, *25*, 109–114.

Laporte, G., & Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, *13*, 133–142.

Li, M., Hao, J.-K., & Wu, Q. (2022). Learning-driven feasible and infeasible tabu search for airport gate assignment. *European Journal of Operational Research*, *302*, 172–186.

Li, Y., Clarke, J.-P., & Dey, S. S. (2021). Using submodularity within column generation to solve the flight-to-gate assignment problem. *Transportation Research Part C: Emerging Technologies*, *129*, 103217.

Liu, M., Liang, B., Zheng, F., Chu, C., & Chu, F. (2018). A Two-stage Stochastic Programming Approach for Aircraft Landing Problem. In *2018 15th International Conference on Service Systems and Service Management (ICSSSM)* (pp. 1–6). ISSN: 2161-1904.

Liu, W., Zhao, Q., & Delahaye, D. (2022). Research on slot allocation for airport network in the presence of uncertainty. *Journal of Air Transport Management*, *104*, 102269.

Liu, Z., & Xiang, Q. (2023). A branch-and-price algorithm for the airport gate assignment problem considering the trade-off between robustness and efficiency. *Transportation Research Part C: Emerging Technologies*, *154*, 104232.

Narciso, M. E., & Piera, M. A. (2015). Robust gate assignment procedures from an airport management perspective. *Omega*, *50*, 82–95.

Okur, G. E., Karsu, Ö., & Solyalı, O. (2025). Lagrangian relaxation for airport gate assignment problem. *Computers & Operations Research*, (p. 107186).

Pellegrini, P., Bolić, T., Castelli, L., & Pesenti, R. (2017). SOSTA: An effective model for the Simultaneous Optimisation of airport SloT Allocation. *Transportation Research Part E: Logistics and Transportation Review*, *99*, 34–53.

Pellegrini, P., Castelli, L., & Pesenti, R. (2012). Metaheuristic algorithms for the simultaneous slot allocation problem. *IET Intelligent Transport Systems*, *6*, 453–462.

Ribeiro, N. A., Jacquillat, A., & Antunes, A. P. (2019). A Large-Scale Neighborhood Search Approach to Airport Slot Allocation. *Transportation Science*, *53*, 1772–1797.

Ribeiro, N. A., Jacquillat, A., Antunes, A. P., Odoni, A. R., & Pita, J. P. (2018). An optimization approach for airport slot allocation under IATA guidelines. *Transportation Research Part B: Methodological*, *112*, 132–156.

Richetta, O., & Odoni, A. R. (1994). Dynamic solution to the ground-holding problem in air traffic control. *Transportation Research Part A: Policy and Practice*, *28*, 167–185.

Şeker, M., & Noyan, N. (2012). Stochastic optimization models for the airport gate assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, *48*, 438–459.

Shone, R., Glazebrook, K., & Zografos, K. G. (2021). Applications of stochastic modeling in air traffic management: Methods, challenges and opportunities for solving air traffic problems under uncertainty. *European Journal of Operational Research*, *292*, 1–26.

Solak, S., Solveling, G., Clarke, J.-P. B., & Johnson, E. L. (2018). Stochastic runway scheduling. *Transportation Science*, *52*, 917–940.

Solveling, G., Solak, S., Clarke, J.-P., & Johnson, E. (2011). Runway Operations Optimization in the Presence of Uncertainties. *Journal of Guidance, Control, and Dynamics*, *34*, 1373–1382.

Wang, D., & Zhao, Q. (2020). A simultaneous optimization model for airport network slot allocation under uncertain capacity. *Sustainability*, *12*, 5512.

Wang, K., & Jacquillat, A. (2020). A Stochastic Integer Programming Approach to Air Traffic Scheduling and Operations. *Operations Research*, *68*, 1375–1402.

Wang, Y., Liu, C., Wang, H., & Duong, V. (2023a). Slot allocation for a multiple-airport system considering airspace capacity and flying time uncertainty. *Transportation Research Part C: Emerging Technologies*, *153*, 104185.

Wang, Y., Wang, M., Xu, W., & Hansen, M. (2023b). Secondary trading of airport slots: Issues and challenges. *Chinese Journal of Aeronautics*, *36*, 1–12.

Yan, S., & Chang, C.-M. (1998). A network model for gate assignment. *Journal of Advanced Transportation*, *32*, 176–189.

Yan, S., & Tang, C.-H. (2007). A heuristic approach for airport gate assignments for stochastic flight delays. *European Journal of Operational Research*, *180*, 547–567.

Yu, C., Zhang, D., & Lau, H. Y. (2017). An adaptive large neighborhood search heuristic for solving a robust gate assignment problem. *Expert Systems with Applications*, *84*, 143–154.

Zeng, W., Ren, Y., Wei, W., & Yang, Z. (2021). A data-driven flight schedule optimization model considering the uncertainty of operational displacement. *Computers & Operations Research*, *133*, 105328.

Zhao, L., & Zeng, B. (2012). An exact algorithm for two-stage robust optimization with mixed integer recourse problems.

Zografos, K. G., Androutsopoulos, K. N., & Madas, M. A. (2018). Minding the gap: Optimizing airport schedule displacement and acceptability. *Transportation Research Part A: Policy and Practice*, *114*, 203–221.

Zografos, K. G., & Jiang, Y. (2019). A Bi-objective efficiency-fairness model for scheduling slots at congested airports. *Transportation Research Part C: Emerging Technologies*, *102*, 336–350.

Zografos, K. G., Madas, M. A., & Androutsopoulos, K. N. (2017). Increasing airport capacity utilisation through optimum slot scheduling: review of current developments and identification of future needs. *Journal of Scheduling*, *20*, 3–24.

Zografos, K. G., Salouras, Y., & Madas, M. A. (2012). Dealing with the efficient allocation of scarce resources at congested airports. *Transportation Research Part C: Emerging Technologies*, *21*, 244–256.