# ENHANCING EFFICIENCY IN AIRSIDE OPERATIONS THROUGH PRESCRIPTIVE ANALYTICS

CHENLIANG ZHANG

PhD

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University

Department of Aeronautical and Aviation Engineering

# Enhancing Efficiency in Airside Operations through Prescriptive Analytics

Chenliang Zhang

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

December 2024

## CERTIFICATE OF ORGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no materials 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)

ZHANG Chenliang (Name of student)

# Abstract

Given the growth in air traffic demand and the highly competitive nature of the aviation industry, many airports face capacity challenges, leading to increasingly frequent and severe congestion. The aviation sector increasingly relies on predictive and optimisation techniques to fully utilise the massive data generated in daily operations. This thesis uses real-world data to provide informed decisions for airside operations by employing advanced prescriptive analytics methodologies, thereby improving efficiency and alleviating congestion.

The first study introduces two prescriptive analytics approaches for the airport gate assignment problem, utilising historical data through machine learning (ML) techniques to enhance decision-making effectiveness and robustness. Supported by effective ML methods and scenario selection strategies, the estimate-then-optimise (ETO) approach delivers superior performance compared to other optimisation techniques. Additionally, we develop an efficient exact solution method, the Benders-based branch-and-cut (BBC) method, to effectively handle real-world scale test instances. This solution method demonstrates statistically significant improvements in computational performance over commercial solvers.

The second study examines the aircraft sequencing and scheduling problem (ASSP) at a single-runway airport under uncertainty in aircraft arrival and departure times. We first introduce an ETO approach, utilising prediction results to drive the stochastic programming model for the ASSP. To address the suboptimal decision-making caused by

prediction errors, we further propose an estimate-then-distributionally-robust-optimise (ETDRO) approach, which incorporates prediction results into a distributionally robust optimisation model for decision-making. Experimental results demonstrate that the ETDRO approach outperforms other optimisation techniques. Additionally, to effectively implement the ETDRO approach, we propose several exact and inexact decomposition methods. Extensive computational results show that our inexact decomposition method can provide optimal or near-optimal solutions for real-world scale test instances within a very short CPU time.

In the third study, we focus on the prescriptive analytics of the multi-runway aircraft landing problem (MALP) with uncertain aircraft arrival times, aiming to design efficient and environmentally friendly landing operations. Following the ETO approach in prescriptive analytics, we employ ML techniques to estimate the distribution of uncertain arrival time based on historical data. An optimisation-enhanced learning-driven scenario generation (OLSG) method is used to generate scenarios that closely resemble actual scenarios based on the estimated distributions, thus preventing the subsequent optimisation from being affected by extreme scenarios and producing suboptimal decisions. Experimental results demonstrate the superior performance of the ETO approach supported by the OLSG method over other optimisation methods. Additionally, we propose a novel exact solution method called the stabilised branch-and-check (SBAC) method to solve the ETO approach for MALP efficiently. This method stabilises the master problem around a neighbourhood of a stable centre point, enabling the generation of strong Benders cuts. The results of computational experiments demonstrate that the proposed SBAC method achieves statistically significant improvements in CPU time compared to the benchmark methods.

This thesis demonstrates the robustness and effectiveness of combining machine learning algorithms, optimisation techniques, uncertainty modelling, and advanced decomposition methods to address key operational challenges in airside operations. The

results include significant improvements in operational efficiency, as well as economic and environmental benefits. In addition, the successful application of prescriptive analytics demonstrates the significant potential and advantages of data-driven decision-making in complex aviation operational environments. This suggests that these data-driven decision-making approaches can be introduced as innovative solutions to address various operational challenges within the aviation industry.

Chief supervisor: Prof. Kam K.H. Ng

Co-supervisor: Prof. Gangyan Xu

# Publications

***: Corresponding author.**

### Arising from the Thesis

#### Journal articles (in reverse chronological order)

1. Zhang, C., Jin, Z., Ng, K.K.H.*, Liu, Y., Wu, L. Stabilised branch-and-check method for optimising multi-runway aircraft landing problem modelled through stochastic programming with learning-driven arrival time predictions. Under first review at *Transportation Research Part E: Logistics and Transportation Review*.

2. Zhang, C., Jin, Z., Ng, K.K.H.*. Tang, T., Tang, R. Distributionally robust optimisation approach for aircraft sequencing and scheduling with learning-driven arrival and departure time predictions. Under first review at *Omega-international journal of management science*.

3. Zhang, C., Ng, K.K.H.*, Jin, Z., Sun, X., Qin, Y. Risk-averse two-stage stochastic programming approach for gate assignment problem under arrival time uncertainty. Under second review at *Computers & Industrial Engineering*.

4. Zhang, C., Ng, K.K.H.*, Jin, Z., Yao, S., Qin, Y. Q-learning-driven exact and meta-heuristic algorithms for the robust gate assignment problem. Under second review at *Advanced Engineering Informatics*.

5. Zhang, C., Jin, Z., Ng, K.K.H.*, Tang, T., Zhang, F., Liu, W., 2025. Predictive and prescriptive analytics for robust airport gate assignment planning in airside

operations under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 195, 103963.

**Others**

**Journal articles (in reverse chronological order)**

1. Jin, Z., Zhang, C., Ng, K.K.H.*, Tian, X., Wang, H., Xu, G. Humanitarian logistics planning under uncertain demand and third-party supply capacity: A distributionally robust optimisation approach. Under Second review at *International Journal of Production Research*.

2. Jin, Z., Ng, K.K.H., Zhang, C.*, Chan, Y., Qin, Y., 2025. A multistage stochastic programming approach for drone-supported humanitarian last-mile logistics system planning. *Advanced Engineering Informatics*, 65, 103201.

3. Jin, Z., Ng, K.K.H., Wang, H.*, Wang, S., Zhang, C., 2024. Electric airport ferry vehicle scheduling problem for sustainable operation. *Journal of Air Transport Management*, 123, 102711.

4. Yeung, T., Zhang, C., Jin, Z., Ng, K.K.H.*, 2024. A joint stochastic programming approach for aircraft landing and terminal traffic flow problem under weather scenarios. *Journal of the Air Transport Research Society*, 3, 100046.

5. Jin, Z., Ng, K.K.H.*, Zhang, C., Liu, W., Zhang, F., Xu, G. 2024. A risk-averse distributionally robust optimisation approach for drone-supported relief facility location problem. *Transportation Research Part E: Logistics and Transportation Review*, 186, 103538.

6. Jin, Z., Ng, K.K.H.*, Zhang, C., Wu, L., Li, A. 2024. Integrated optimisation of strategic planning and service operations for urban air mobility systems. *Transportation Research Part A: Policy and Practice*, 183, 104059.

7. Jin, Z., Ng, K.K.H.*, Zhang, C., 2024. Robust optimisation for vertiport location problem considering travel mode choice behaviour in urban air mobility systems. *Journal of the Air Transport Research Society*, 2, 100006.

**Conference Articles**

1. Zhang, C., Jin, Z., Ng, K.K.H.* (2024, Nov). Distributionally robust optimisation approach for aircraft sequencing and scheduling with learning-driven arrival and departure time predictions. *2024 Air Transport Research Society China Chapter. Beijing, China.*

2. Jin, Z., Zhang, C., Ng, K.K.H.*, (2024, Jul). An improved variable neighbourhood search for the gate assignment problem with time windows. *AIAA Aviation Forum. Las Vegas, Nevada, U.S.*

3. Jin, Z., Ng, K.K.H.*, Zhang, C., (2023, Jul). Robust optimisation for the vertiport location problem in urban air mobility system. *The 26th ATRS World Conference. Kobe, Japan.*

4. Zhang, C., Ng, K.K.H.*, (2022, Jul). A variable neighbourhood search for the dynamic gate assignment problem with time windows. *The 25th ATRS World Conference. Antwerp, Belgium.*

# Contents

# Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Kam K.H. Ng, for his continuous and invaluable support over the past three years. His guidance has been indispensable not only in my academic research but also in all aspects of my life. Whenever I encounter difficulties in my research, Dr. Ng is always willing to help. This thesis would not have been possible without his support. Dr. Ng's dedication to research, passion for work, and great patience in teaching have continually inspired me to make consistent progress. From him, I have learnt how to be an excellent researcher and a better person. His guidance has profoundly impacted my future academic and personal endeavours.

Special thanks go to my thesis examiners: Dr. Li-Ta Hsu from PolyU, Prof. Gabriel Lodewijks from the University of Newcastle, and Dr. Peng Wei from George Washington University. I am grateful for their precious time and insightful comments on my thesis.

Heartfelt thanks are also due to my co-supervisor, Dr. Gangyan Xu from PolyU, and my former supervisors, Dr. Fangni Zhang from HKU and Prof. Chong Ye from FZU. Their guidance and encouragement have continually inspired me to advance my research efforts. I would also like to thank Dr. Wei Liu, Dr. Qinbiao Li, Dr. Lingxiao Wu, Prof. Tie-qiao Tang, Dr. Yichen Qin, Dr. Xuting Sun, and many others for their lectures and guidance, through which I acquired essential knowledge and skills in conducting research. I am also grateful to the staff of AAE for their assistance in both

my studies and life at PolyU.

I sincerely thank my outstanding collaborator, Zhongyi, for his significant assistance and professional cooperation throughout my thesis and other research papers, which enabled me to complete my work successfully. Thanks to the other members of the Air Traffic Management Research Group: Xin, Chiu, James, Nana, Ye, Senna, Dave, Alan, Yipeng, Congcong, Changxin, Rong, Qinyu, Yuanyuan, and Terrence, for their support and collaboration. Also, thanks to my friends Canbin, Mengtao, Songyi, Zhuo, Haoqing, Haixuan, Jingxuan, Jiahao, and Jingfeng.

Finally, I would like to dedicate this thesis to my parents and aunt for their endless love and support.

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| AGAP | Airport gate assignment problem |
| AGRP | Aircraft ground routing problem |
| ALP | Aircraft landing problem |
| ANN | Artificial neural networks |
| ASSP | Aircraft sequencing and scheduling problem |
| ATC | Air traffic control |
| ATM | Air traffic management |
| ATP | Aircraft take-off problem |
| BAC | Branch-and-check |
| BBC | Benders-based branch-and-cut |
| BD | Benders decomposition |
| BOM | Chhatrapati Shivaji Maharaj International Airport |
| CDG | Charles-de-Gaulle |
| CPS | Constrained position shifting |
| CSR | Clustered-based scenario reduction |
| CVaR | Conditional Value at Risk |
| DFJ | Dantzig-Fulkerson-Johnson |
| DR L-shaped | Distributionally robust L-shaped |
| DRO | Distributionally robust optimise |
| DXB | Dubai International Airport |

| | |
|---|---|
| EALP | Extended aircraft landing problem |
| E-DR-BBC | Enhanced distributionally robust Benders-based branch-and-cut |
| E-DR L-shaped | Enhanced distributionally robust L-shaped |
| EEV | Expected value of the expected value solution |
| ETDRO | Estimate-then-distributionally-robust-optimise |
| ETO | Estimate-then-optimise |
| FL | Federated learning |
| FUK | Fukuoka Airport |
| HDSG | Historical data-driven scenario generation |
| HKIA | Hong Kong International Airport |
| IAF | Initial approach fix |
| ICAO | International Civil Aviation Organisation |
| IQR | Interquartile Range |
| KNN | K-nearest neighbour |
| LAX | Los Angeles International Airport |
| LB | Lower bound |
| LBBD | Logic-based Benders decomposition |
| LBL | Lower bound lifting |
| LGW | London Gatwick Airport |
| LHR | London Heathrow Airport |
| LP | Linear programming |
| LSG | Learning-driven scenario generation |
| MAE | Mean absolute error |
| MBE | Mean bias error |
| MALP | Multi-runway aircraft landing problem |
| MILP | Mixed-integer linear programming |

| | |
|---|---|
| MIP | Mixed-integer programming |
| ML | Machine learning |
| MSE | Mean square error |
| OLSG | Optimisation-enhanced learning-driven scenario generation |
| PTO | Predict-then-optimise |
| PVG | Shanghai Pudong International Airport |
| RF | Random forest |
| RMSE | Root mean square error |
| RO | Robust optimisation |
| RSP | Runway scheduling problem |
| SAA | Sample average approximation |
| SAN | San Diego International Airport |
| SBAC | Stabilised branch-and-check |
| S-DR-BBC | Stabilised distributionally robust Benders-based branch-and-cut |
| SFHD | Sampling from the historical data |
| SP | Stochastic programming |
| SP-CR | Stochastic programming with continuous recourse |
| SP-MIR | Stochastic programming with mixed-integer recourse |
| STN | London Stansted Airport |
| TAF | Terminal area forecast |
| UB | Upper bound |
| URC | Urumqi Diwopu International Airport |
| VSS | Value of the stochastic solution |
| XMN | Xiamen Gaoqi International Airport |
| XGBoost | eXtreme gradient boosting |

# Chapter 1

# Introduction

## 1.1 Background

Given the growth in air traffic demand and the highly competitive nature of the airline industry, many airports face capacity challenges, resulting in increasingly frequent and severe congestion (Ng et al., 2017; Solak et al., 2018; Ribeiro et al., 2019; Bi et al., 2022). In a constrained capacity environment, any reduction in system capacity can lead to significant delays and substantial losses for airlines and passengers, posing severe challenges to the air traffic management (ATM) system (Solak et al., 2018; Khassiba et al., 2020). Due to the high investment costs and prolonged construction periods, constructing new infrastructure, such as airports, terminals, runways, and taxiways, to increase the capacity of airspace systems is often not an immediate solution (Ikli et al., 2021). Consequently, there is a pressing need to optimise the utilisation of existing airside facilities to enhance the efficiency of airside operations and alleviate congestion (Bouras et al., 2014; Daş et al., 2020; Wang et al., 2022a; Chen et al., 2024). Optimisation problems in airside operations primarily encompass the airport gate assignment problem (AGAP) (Bouras et al., 2014; Daş et al., 2020), runway scheduling problem (RSP) (Bennell et al., 2011; Ikli et al., 2021), and aircraft ground routing problem

(AGRP) (Guépet et al., 2016), etc.

Deterministic optimisation problems for airside operations, which assume that all input information is known with certainty, have been extensively studied (Bennell et al., 2011; Bouras et al., 2014; Guépet et al., 2016; Daş et al., 2020; Ikli et al., 2021; Messaoud, 2021). However, with increasing air traffic and factors such as severe weather, delay propagation, the probabilistic nature of trajectories, technical difficulties, and security concerns, the aircraft arrival and departure times have become increasingly uncertain (Şeker and Noyan, 2012; Ng et al., 2017; Solak et al., 2018; Khassiba et al., 2020; Kim et al., 2023a). Uncertain parameters can render predetermined airside operation plans ineffective. Consequently, developing robust plans to address these uncertainties has become a key focus in recent research on airside operations (Şeker and Noyan, 2012; Ng et al., 2017; Xu et al., 2017; Solak et al., 2018; Brownlee et al., 2018; Khassiba et al., 2020; Wang et al., 2021; Khassiba et al., 2022; Kim et al., 2023a).

Stochastic optimisation methods, including stochastic programming (SP) (Şeker and Noyan, 2012; Solak et al., 2018; Khassiba et al., 2020, 2022) and robust optimisation (RO) (Solveling et al., 2011; Ng et al., 2017; Xu et al., 2017), are widely employed for the robust and efficient planning of airside operations. In the aforementioned uncertainty modelling and optimisation approaches for airside operations, the distributions of uncertain arrival and departure times are primarily derived from the analysis of historical data (Solveling et al., 2011; Xu et al., 2017; Solak et al., 2018) or empirical knowledge (Ng et al., 2017; Khassiba et al., 2020, 2022). With the advancement of big data technology, airports have accumulated massive airside operation data. By analysing and mining this data, airports can gain deeper insights into airside operations, identify potential issues, and optimise operational processes. Additionally, the extensive airside operation data presents new opportunities for addressing uncertainty in optimisation problems related to airside operations. The combination of predictive and optimisation techniques to make informed decisions based on available data that

can be used to predict uncertain parameters is known as prescriptive analytics (Bertsimas and Kallus, 2020; Qi and Shen, 2022; Wang and Yan, 2023; Tian et al., 2023a,b,c). Specifically, the available data can be categorised into historical data of the uncertain parameters and other auxiliary data that can be used to predict.

Although prescriptive analytics approaches have been introduced in maritime and land transportation research (Yan et al., 2020; Kandula et al., 2021; Tian et al., 2023a,b; Luo et al., 2023; Yan et al., 2024a), such approaches are still rarely applied in air transportation. This thesis introduces the prescriptive analytics approaches to various airside operations. The integration of predictive and optimisation methods, supported by extensive airside operation data, enables informed decision-making for airside operations under conditions of uncertainty. However, prescriptive analytics approaches typically require substantial information from predictions to be input into the subsequent optimisation problems, often increasing their complexity. To address this issue, we further develop effective scenario selection strategies tailored to different prediction performances, as well as high-performance solution methods for various optimisation problem structures to enhance scalability and computational tractability. These methods make prescriptive analytics approaches applicable to real-world airside operations.

## 1.2 Research scope and objectives

This thesis addresses operations management problems in the airside sector under conditions of uncertainty, aiming to develop robust solutions. By integrating predictive and optimisation methods, referred to as prescriptive analytics, we can make informed decisions for airside operations based on the available data. However, these prescriptive analytics approaches often increase problem complexity. To address this, we develop high-performance solution methods tailored to the specific structure of different problems, enhancing scalability and computational tractability. This thesis comprises three

studies:

In the first study, we examine prescriptive analytics for the AGAP. Two prescriptive analytics approaches are employed to develop robust and efficient airport gate assignment plans. Initially, we use a predict-then-optimise (PTO) approach, utilising machine learning (ML) methods to predict aircraft arrival times, thereby providing more accurate information for the AGAP. Subsequently, we explore an estimate-then-optimise (ETO) approach. This approach begins with estimating the conditional distribution of uncertain aircraft arrival times using ML, followed by solving an SP model for the AGAP based on the estimated distribution. This SP model involves two decision stages: the first stage assigns aircraft to either contact gates or the apron and determines the sequence of aircraft for each contact gate; the second stage designs aircraft scheduling plans based on the observed arrival times. Due to the complexity of solving the ETO approach, we develop an effective scenario selection strategy and an efficient Benders-based branch-and-cut (BBC) method.

In the second study, we investigate prescriptive analytics for the aircraft sequencing and scheduling problem (ASSP). We first introduce an ETO approach to address the ASSP under uncertainties in landing and take-off times on a mixed-operation single runway. The ETO approach, a type of prescriptive analytics, involves two steps. First, the ML method predicts the distribution of aircraft landing and take-off times. Second, a SP model for the ASSP is solved based on the estimated distribution. In real-world applications, predictive methods may produce forecasts susceptible to errors or distribution shifts. To ensure robust aircraft sequencing and scheduling decisions despite misspecified distributional information, we replace the SP model in the second step of the ETO approach with distributionally robust optimisation (DRO), thereby proposing the estimate-then-distributionally-robust-optimise (ETDRO) approach for the ASSP. We develop several exact and inexact decomposition methods to handle the ETDRO approach for the ASSP, enhancing them with lower bound lifting cuts to

tighten the lower bounds and a heuristic to identify high-quality upper bounds.

In the third study, we focus on prescriptive analytics for the multi-runway aircraft landing problem (MALP) under aircraft arrival time uncertainty, aiming to devise efficient and environmentally friendly landing operations. We formulate the problem as an SP model with a mixed-integer recourse (SP-MIR). Adhering to the ETO framework in prescriptive analytics, we employ ML techniques to estimate the distribution of unknown parameters based on historical data. An optimisation-based scenario generation method is utilised to create scenarios that closely resemble actual scenarios, thereby avoiding decision biases influenced by extreme scenarios. Furthermore, we introduce a novel exact method called the stabilised branch-and-check (SBAC) method to solve the ETO approach for the MALP efficiently. This method stabilises the master problem around a neighbourhood of a stability centre point, enabling the generation of strong Benders cuts.

## 1.3 Thesis outline

The remainder of the thesis is organised as follows. Chapter 2 summarises and reviews existing literature on airside operations under uncertainty, and prescriptive analytics that combine optimisation with prediction for informed decision-making. Chapter 3 develops ML methods for predicting aircraft arrival times by incorporating data on aircraft operations and aviation meteorological conditions. These predictions are fed into the AGAP model to provide aircraft-to-gate assignments at the airside by employing PTO and ETO methods. Chapter 4 introduces an RF model for predicting aircraft arrival and departure times, with these predictions serving as input to the ASSP model for single runway operations. The ETDRO approach based on the Wasserstein distance is employed to handle prediction and sampling errors in aircraft sequencing and scheduling decision-making. Chapter 5 presents an OLSG method to make high-quality

decisions by selecting appropriate scenarios from ML predictions. These generated scenarios are then input into the SP-MIR model for the MALP to develop efficient and robust aircraft landing plans for a multi-runway system. Chapter 6 concludes the thesis and discusses some future research directions.

# Chapter 2

# Literature review

This chapter illustrates the literature review on relevant research concerning airside operations under uncertainty, as well as the prescriptive analytics that combines optimisation and prediction for informed decision-making.

## 2.1 Airside operations under uncertainty

This section reviews the existing literature on the AGAP and the RSP under uncertainty, as investigated in this thesis.

### 2.1.1 AGAP under uncertainty

The deterministic AGAP has been extensively studied in recent years due to its practical importance (Daş et al., 2020; Li et al., 2021; Karsu et al., 2021; Wang et al., 2022a; Karsu and Solyalı, 2023). The generic form of the AGAP requires two primary constraints to be satisfied: first, each aircraft must be assigned to a gate or apron; second, aircraft assignments with overlapping ground times at the same gate must be prevented (Daş et al., 2020). Recent studies have incorporated space restrictions and gate compatibility constraints in the AGAP models (Daş et al., 2020; Li et al., 2021;

Bi et al., 2022). The objectives of the AGAP are mainly oriented towards passengers, airports, and airlines (Daş et al., 2020). Passenger-oriented objectives typically aim to minimise walking distance, waiting time, transit time, and other factors that affect passenger satisfaction and convenience (Daş, 2017; Karsu et al., 2021). Airport- and airline-oriented objectives focus on assigning fewer aircraft to aprons or more aircraft to preferred gates (Dorndorf et al., 2008; Dell'Orco et al., 2017; Daş et al., 2020). Additionally, robustness-oriented objective functions are employed in the deterministic AGAP to provide robust gate assignment plans to address uncertainties. These functions include minimising the variance of idle times (Deng et al., 2017; Wang et al., 2022a), maximising idle times between consecutive aircraft (Dorndorf et al., 2008; Benlic et al., 2017), minimising the expected number and duration of gate conflicts (Castaing et al., 2016; Yu et al., 2017), and minimising aircraft delay times (Li et al., 2021). Furthermore, several studies consider the trade-off between operational efficiency and robustness as objectives (Benlic et al., 2017; Daş, 2017; Liu and Xiang, 2023).

The deterministic AGAP with robustness-oriented objectives does not guarantee that the established airport gate assignment plans can be executed as expected for most scenarios on an operational day (Şeker and Noyan, 2012; Kim et al., 2023a). Consequently, uncertainty modelling and optimisation approaches are utilised for the AGAP. Şeker and Noyan (2012) employed a scenario-based SP approach for the AGAP, utilising a two-stage decision-making process. They used various robustness measures, such as the number of conflicting flights, idle times, and buffer times, as objective functions in different models. Scenarios were generated by adding deviations to the estimated aircraft arrival times, with the deviation amounts randomly derived from a predetermined triangular distribution. A Tabu search heuristic was designed to obtain acceptable solutions for the proposed AGAP model within a reasonable time frame. Xu et al. (2017) introduced an RO approach for the AGAP using a two-stage decision-making process. Their model aims to minimise a predetermined quantile of total over-

lap time. To address the robust counterpart, they incorporated a solution-dependent uncertainty budget. Upon analysing real data, they observed that the distribution of aircraft delays exhibited a long tail due to extreme values. After removing the outliers, the distribution became more balanced and symmetrical. Kim et al. (2023a) proposed an overlap chance-constrained AGAP that aims to balance robustness and efficiency with a specified probability limiting the occurrences of overlap. The distribution of aircraft overlap probabilities is generated using historical aircraft deviation data.

### 2.1.2 RSP under uncertainty

The optimisation problem aimed at improving runway utilisation is known as the RSP (Bennell et al., 2011; Ikli et al., 2021). It can be further divided into three categories: the aircraft landing problem (ALP) for arriving aircraft, the aircraft take-off problem (ATP) for departing aircraft, and the ASSP for both landing and taking-off aircraft (Bennell et al., 2011; Ng et al., 2017; Ikli et al., 2021; Messaoud, 2021). The RSP determines the sequence and schedule of landing and taking-off aircraft to optimise predefined objectives while adhering to various operational constraints. These constraints generally include spacing, wake vortex separation requirements, and operational time windows (Bennell et al., 2011; Solak et al., 2018; Ikli et al., 2021). The literature discusses two main objective functions for the RSP: minimising the makespan of the runway system (Balakrishnan and Chandran, 2010; Harikiopoulo and Neogi, 2010; Prakash et al., 2021) and minimising the total, average, or weighted delay of all aircraft (Sama et al., 2017; Pohl et al., 2021; Prakash et al., 2022; Pohl et al., 2022).

The deterministic RSP, where all input information is assumed to be deterministic, has been extensively studied (Bennell et al., 2011; Solak et al., 2018; Ikli et al., 2021; Messaoud, 2021). However, with increasing air traffic and factors such as severe weather, delay propagation, the probabilistic nature of trajectories, technical difficulties, and security concerns, aircraft arrival and departure times have become increas-

ingly uncertain (Ng et al., 2017; Solak et al., 2018; Khassiba et al., 2020, 2022; Kim et al., 2023a). These uncertainties can render predetermined runway scheduling plans ineffective. Consequently, developing robust plans to address these uncertainties has become a key focus in recent runway operations research.

Solveling et al. (2011) developed an SP approach for the ASSP under landing and take-off time uncertainty in a parallel runway system, with one runway dedicated to landings and the other to take-offs. Their results demonstrate the potential benefits of the SP model over a deterministic model during peak hours. Ng et al. (2017) investigated the ASSP under uncertainty in a mixed-mode parallel runway system, employing an RO approach to develop aircraft sequencing and scheduling plans for worst-case scenarios. Solak et al. (2018) introduced SP models based on network and slot formulations for the ASSP under uncertainty, conducting experiments in a parallel runway system with two independently operated runways, one for landing and one for take-off. The results indicate that the proposed SP models are practically implementable and offer potential advantages over deterministic models. Khassiba et al. (2020) proposed an SP model with chance constraints for the extended aircraft landing problem (EALP) on a single landing runway, aiming to pre-schedule aircraft at a destination airport to minimise landing sequence length and time-deviation costs. Validation with realistic data from Charles-de-Gaulle (CDG) airport demonstrates its advantages over deterministic policies. Khassiba et al. (2022) extended previous research Khassiba et al. (2020) on the EALP under uncertainty by incorporating multiple initial approach fixes (IAFs) and different initial aircraft statuses. The study introduces two problem variants based on IAF assignment flexibility, using realistic data from CDG airport to demonstrate the benefits of the SP models and IAF re-assignment.

### 2.1.3 Research gaps

Based on the review of previous studies on the AGAP and RSP under uncertainty, we have identified several research gaps, which are discussed in this subsection.

The first research gap is that in previous studies of AGAP and RSP under uncertainty, the distributions of uncertain parameters primarily originate from experiential knowledge (Şeker and Noyan, 2012; Ng et al., 2017; Khassiba et al., 2020, 2022) or the historical data (Solveling et al., 2011; Xu et al., 2017; Solak et al., 2018; Kim et al., 2023a). With the development of big data technology in the aviation industry, airside controllers can access additional information beyond historical data on uncertain parameters, enabling ML methods to estimate their distributions accurately. Driven by these predictive results, optimisation approaches can generate airport gate assignment plans and aircraft sequencing and scheduling plans closer to the actual scenarios on the operational days.

The second research gap is that previous studies on the AGAP and RSP under uncertainty have mainly utilised SP and RO approaches. The SP approach requires precise distributional information about uncertain parameters, which is often challenging to obtain. In contrast, RO aims to optimise performance under the worst-case scenario, potentially resulting in conservative solutions. The DRO approach offers a promising alternative to address the limitations of these two approaches. Unlike the SP approach, the DRO approach does not require complete distributional information about uncertain parameters. Additionally, the DRO approach optimises performance across various possible distributions, resulting in less conservative solutions than the RO approach (Delage and Ye, 2010; Bertsimas et al., 2019).

The final research gap is that previous studies addressing runway operations under uncertainty primarily focus on single runways or dual runways that handle aircraft take-offs and landings independently. However, due to substantial air traffic, many

major airports worldwide now feature multiple runways dedicated to aircraft landing operations (Messaoud, 2021). Consequently, we further explore the MALP under uncertainty. The aim of the MALP under uncertainty is to devise aircraft landing plans that enable efficient landing operations on a multi-runway system. This model can also be easily adapted to the multi-runway aircraft take-off problem. Besides, air traffic control (ATC) must make decisions regarding aircraft assignment, sequencing, and scheduling within a parallel multi-runway system for aircraft landing operations. ATC typically determines aircraft-to-runway assignment before the aircraft enters the terminal airspace. Aircraft sequencing decisions are usually made when the aircraft is in the terminal airspace, while scheduling decisions are typically made when the aircraft enters the final approach phase. In the previous SP model for runway operations, decisions are based on a two-stage process (Solak et al., 2018; Khassiba et al., 2020). These studies often assume that aircraft arrival times are known when the aircraft enters the final approach phase. Consequently, aircraft sequencing decisions are usually addressed in the first stage, while aircraft scheduling decisions are handled in the second stage after the arrival times of aircraft are revealed. Since the second-stage problem of this SP model, which considers only the aircraft scheduling decision, is formulated as a linear programming (LP) problem with continuous variables, we refer to this SP model as the SP model with continuous recourse (SP-CR). With the support of advanced aviation technologies, including precise navigation systems, real-time data processing, accurate predictive analysis, and efficient communication systems, ATC can monitor the flight status of aircraft in real-time and make relatively accurate arrival time predictions when aircraft are operating in terminal airspace. Consequently, ATC can typically make aircraft sequencing decisions based on these revealed arrival times. Therefore, we can incorporate aircraft sequencing decisions into the second stage of the SP model, thereby proposing an SP-MIR model for the MALP. This model assigns arriving aircraft to runways in the first stage. Subsequently, in the second stage, it

makes sequencing and scheduling decisions for the aircraft assigned to each runway using a MILP model.

## 2.2 Prescriptive analytics

The combination of predictive and optimisation methods to make informed decisions based on available data is known as prescriptive analytics (Bertsimas and Kallus, 2020; Qi and Shen, 2022; Wang and Yan, 2023; Tian et al., 2023a,b,c). Prescriptive analytics approaches are generally divided into indirect and direct paths based on the relationship between prediction and optimisation methods (Tian et al., 2023c). The indirect path involves first obtaining predictions or estimates through predictive analysis, which are subsequently used as inputs to downstream optimisation problems. This path typically includes the PTO and the ETO approaches (Yan et al., 2022; Luo et al., 2023; Wang et al., 2023; Yang et al., 2024; Yan et al., 2024b; Tian et al., 2023c). The direct path leverages the structure of the optimisation problem (i.e., its objectives and constraints) to design better prediction models, aiming to transition directly from data to decision (Elmachtoub and Grigas, 2022; Tian et al., 2023a,c). This path typically includes smart predict-then-optimise, weighted sample average approximation, empirical risk minimisation, and kernel optimisation approaches (Bertsimas and Kallus, 2020; Brandt et al., 2022; Notz and Pibernik, 2022; Elmachtoub and Grigas, 2022; Tian et al., 2023a; Wang et al., 2024).

The direct path leverages the structure of the optimisation problem to design a better prediction model. This approach prioritises decision quality but can involve significant computational effort. Consequently, the associated optimisation problem is typically required to be relatively simple (Elmachtoub and Grigas, 2022; Tian et al., 2023c). The AGAP and the RSP exhibit NP-hard characteristics, making them complex to solve (Bennell et al., 2011; Bouras et al., 2014; Daş et al., 2020; Ikli et al., 2021).

Therefore, combining these optimisation problems with the direct path in prescriptive analytics approaches might not be appropriate. This thesis focuses on applying the indirect path in prescriptive analytics approaches to handle AGAP and RSP under uncertainty. Although PTO and ETO approaches included in the indirect path are easy to understand and implement, achieving a good combination of prediction and optimisation methods still requires careful design. In this section, we review the relevant literature on PTO and ETO, respectively, and summarise some gaps in current research.

### 2.2.1 PTO approach

The PTO approach is widely adopted because it can easily integrate prediction results with various optimisation problems. In particular, this approach can be viewed as first predicting the best scenario and then optimising the deterministic model using the predicted best scenario as input (Keutchayan et al., 2023). The predict-then-optimise approach is widely applied to transportation optimisation problems and is supported by various machine learning methods such as K-nearest neighbour (KNN), eXtreme gradient boosting (XGBoost), federated learning (FL), random forest (RF), and artificial neural networks (ANN) (Luo et al., 2023; Wang et al., 2023; Yang et al., 2024; Yan et al., 2024a,b). Nevertheless, the PTO approach may lead to suboptimal decisions, especially when the objective function is nonlinear with respect to the predicted values of the uncertain parameters (Qi and Shen, 2022; Yang et al., 2024).

### 2.2.2 ETO approach

Within the ETO approach, the distributions of uncertain parameters are first estimated. Subsequently, an SP model is solved based on these estimated distributions (Bertsimas and Kallus, 2020; Qi and Shen, 2022; Yan et al., 2022; Yang et al., 2024;

Wang et al., 2024). Although the ETO approach can usually provide more satisfactory decisions than the PTO approach, it still requires a well-designed framework to properly incorporate the distribution of the estimated uncertain parameters into the subsequent SP models. In the study of Yan et al. (2022) and Yang et al. (2024), KNN-based ETO approaches were developed, demonstrating their effectiveness in supporting efficient decision-making. In the KNN-based ETO approaches (Yan et al., 2022; Yang et al., 2024), the Cartesian product of neighbourhood sets is used to approximate the distribution of uncertain parameters. The sample average approximation (SAA) method is then employed as a scenario selection strategy to sample an appropriate number of scenarios from the Cartesian product, ensuring the tractability of the subsequent SP model.

### 2.2.3 Research gaps

The first research gap is that the current ETO approach primarily relies on the KNN method (Yan et al., 2022; Yang et al., 2024). However, considering that the KNN method is typically sensitive to noise and outliers in the dataset, these outliers may affect the determination of the nearest neighbours, causing the estimated distribution to deviate significantly from the true distribution, leading to unsatisfactory gate assignment plans. Given that the RF method can handle noise and outliers in the data by combining the prediction results of multiple decision trees, it offers strong robustness and accuracy. Therefore, we further propose an RF-based ETO approach to address this research gap.

The second research gap identified is the prevalent use of the SAA method as a scenario selection strategy in current research (Yan et al., 2022; Yang et al., 2024). Although the SAA method is simple and easy to understand, its results may be biased if the sample size is insufficient. Additionally, for complex optimisation problems, increasing the sample size raises computational complexity. Therefore, adopting an

effective scenario selection strategy to extract an appropriate number of scenarios and their associated probabilities from the Cartesian product is crucial for making satisfactory decisions (Yan et al., 2022; Yang et al., 2024). Therefore, we propose cluster-based and optimisation-enhanced scenario generation methods to achieve better performance than the SAA method.

The last research gap identified is that in the ETO approach, the distribution of uncertain parameters provided by ML methods is assumed to be the true distributional information and is directly used in subsequent SP models. However, scenarios generated by ML methods inherently produce some prediction and sampling errors, which cannot be entirely eliminated. Consequently, ignoring these potential prediction and sampling errors may lead to unsatisfactory planning. We propose a novel ETDRO approach to address this issue. This approach replaces the SP model with a DRO model. The true distribution is assumed to be either completely unknown or only partially known. In contrast, the estimated distribution provided by the ML methods is used as a known reference distribution in the DRO model.

# Chapter 3

# Prescriptive analytics for the airport gate assignment problem

## 3.1 Introduction

Air traffic demand has continued to grow in recent years, and the competitive nature of the airline industry has intensified, leading to significant capacity challenges for many airports (Ribeiro et al., 2019; Bi et al., 2022). Consequently, there is a pressing need to optimise the utilisation of existing airside facilities to improve the efficiency of airside operations and alleviate congestion (Bouras et al., 2014; Daş et al., 2020; Wang et al., 2022a; Chen et al., 2024). Airport gates, including terminal gates and aprons, are critical as essential airside facilities for passenger boarding and disembarking. For this reason, the AGAP, which focuses on the assignment of aircraft to airport gates, has been widely studied in recent years (Bouras et al., 2014; Daş et al., 2020; Karsu et al., 2021; Li et al., 2021; Wang et al., 2022a; Jiang et al., 2023; Liu and Xiang, 2023; Karsu and Solyalı, 2023; Jiang et al., 2024; Li et al., 2024; Nikolić et al., 2024). In these studies, the AGAP is usually developed based on the given estimated aircraft arrival and departure times. Two constraints are required to ensure the feasibility of airport

gate assignment plans. First, each aircraft should be assigned to a compatible gate or apron. Second, it is essential to prevent scheduling conflicts for aircraft assigned to the same gate. However, in actual operations, because of the bad weather, air traffic delay propagation, technical challenges, safety considerations, etc., the estimated aircraft arrival and departure times are not certain (Şeker and Noyan, 2012; Ng et al., 2017; Solak et al., 2018; Khassiba et al., 2020, 2022; Kim et al., 2023a). Such uncertain parameters may render the predetermined airport gate assignment plans unsatisfactory. Therefore, developing robust airport gate assignment plans to address uncertainties has emerged as a key research focus of the AGAP.

Various robustness criteria have been incorporated into the AGAP models to obtain robust airport gate assignment plans (Daş et al., 2020). For example, minimising the variance of idle times (Bolat, 2001; Deng et al., 2017; Wang et al., 2022a), maximising idle times (Benlic et al., 2017; Liu and Xiang, 2023), avoiding the assignment of two aircraft with low idle times to the same gate (Dorndorf, Ulrich and Jaehn, Florian and Pesch, Erwin, 2017), minimising the expected number and duration of gate conflicts (Castaing et al., 2016; Yu et al., 2017; Dorndorf, Ulrich and Jaehn, Florian and Pesch, Erwin, 2017), minimising absolute deviation of new airport gate assignment plans from the reference schedules (Dorndorf et al., 2012; She et al., 2022), etc. However, the robustness-oriented objectives with estimated aircraft arrival and departure times do not guarantee that the established airport gate assignment plans can be executed as expected for most scenarios on the operational day (Şeker and Noyan, 2012; Kim et al., 2023a). Consequently, uncertainty modelling and optimisation approaches are utilised in the AGAP. Şeker and Noyan (2012) used a scenario-based SP approach for AGAP to minimise the expected variance of the idle times and the expected overlap times. The scenarios are generated by adding deviations to the estimated aircraft arrival times, where the deviation amounts are randomly derived from a predetermined triangular distribution. Xu et al. (2017) introduced an RO approach for AGAP, which aims to

minimise a predetermined quantile of the total overlap time. To address the robust counterpart, they introduced a solution-dependent uncertainty budget. Upon analysing the actual data, they observed that the distribution of aircraft delays exhibited a long tail due to extreme values. After removing the outliers, the distribution became more balanced and symmetrical. Kim et al. (2023a) proposed an overlap chance-constrained AGAP that aims to balance robustness and efficiency, where a specified probability limits the occurrences of overlap. The distribution of aircraft overlap probabilities is generated using historical aircraft deviation data.

In previous AGAP studies under uncertainty, the distributions of uncertain parameters primarily originate from experiential knowledge (Şeker and Noyan, 2012) or the analysis of historical data related to these parameters (Xu et al., 2017; Kim et al., 2023a). With the advancement of big data technology, the substantial volume of collected data provides new opportunities to address uncertainty in optimisation problems, which allows for various ML methods to predict uncertain parameters using their historical data and other relevant data (Wang and Yan, 2023; Tian et al., 2023c). The combinations of predictive and optimisation techniques to make informed decisions based on available data are known as prescriptive analytics (Bertsimas and Kallus, 2020; Qi and Shen, 2022; Wang and Yan, 2023; Tian et al., 2023a,b,c). The PTO and ETO approaches are commonly employed in prescriptive analytics to derive decisions from data. Specifically, these frameworks derive predictions or estimations through ML methods and then use the predicted values or the estimated distributions as inputs to downstream decision-making processes. The PTO approach is widely adopted because it can easily integrate prediction results with various optimisation problems. In particular, this approach can be viewed as first predicting the best scenario and then optimising the deterministic model using the predicted best scenario as input (Keutchayan et al., 2023). The PTO approach is widely applied in transportation optimisation problems, supported by ML methods such as KNN, XGBoost, FL,

RF, ANN (Luo et al., 2023; Wang et al., 2023; Yang et al., 2024; Yan et al., 2024a,b). Nevertheless, the PTO approach may lead to suboptimal decisions, especially when the objective function is nonlinear with the predicted values of the uncertain parameters (Qi and Shen, 2022). The ETO approach can be utilised to address this issue. Within this approach, the distributions of uncertain parameters are first estimated, and then an SP model is solved based on the estimated distributions (Qi and Shen, 2022; Yan et al., 2022; Yang et al., 2024; Wang et al., 2024). The ETO approach necessitates a well-designed framework to appropriately incorporate the distribution of the estimated uncertain parameters into the SP model. In the study of Yang et al. (2024), a KNN-based ETO approach was developed, demonstrating its effectiveness in supporting efficient decision-making.

In this study, we implement prescriptive analytics to enhance the efficiency of airport gate assignment planning. By leveraging ML methods and massive data related to uncertain parameters, more accurate predicted values and estimated distributions can be achieved. These prediction results then drive optimisation methods to generate airport gate assignment plans closer to actual scenarios on an operational day. We first apply the PTO approach to the AGAP, where the uncertain parameters in the AGAP are assumed to be identical to the corresponding predicted values provided by ML methods. Subsequently, the ETO approach for AGAP is developed, utilising the distribution of uncertain parameters generated by ML methods as input to solve the SP model of the AGAP. In practical airport gate assignment operations, the gate controllers typically establish the aircraft assignment and sequencing decisions several hours or a day in advance. After the aircraft arrives at the airport, the actual arrival time of each aircraft is revealed, and the airport gate controllers subsequently make aircraft scheduling decisions for each gate. Therefore, we propose an SP model with a two-stage decision-making process for AGAP. In the first stage, aircraft are assigned to appropriate gates, and aircraft sequencing decisions are made for each gate. Then,

in the second stage, aircraft scheduling decisions are determined after the uncertainty of the aircraft arrival times is revealed.

The KNN and RF methods are used to drive the prescriptive analytics in this study. The KNN method is a non-parametric supervised learning method (Peterson, 2009). Due to its ease of use and good performance, the KNN method is widely used in the research of prescriptive analytics (Bertsimas and Kallus, 2020; Galli et al., 2021; Yan et al., 2022; Yang et al., 2024). The RF method is a supervised learning method, which uses multiple decision trees for prediction (Breiman, 2001). Given its advantages, including high prediction accuracy, robustness to outliers, and relatively good interpretability, etc., the RF method is also widely employed in the research of prescriptive analytics (Bertsimas and Kallus, 2020; Yan et al., 2020; Galli et al., 2021; Yan et al., 2024a). Recall that the ETO approach in prescriptive analytics usually requires a well-designed framework to properly incorporate the distribution of the estimated uncertain parameters into the SP model. Yang et al. (2024) proposed a KNN-based ETO approach. However, considering that the KNN method is usually sensitive to noise and outliers in the dataset, these outliers may affect the judgment of the nearest neighbours, causing the estimated distribution to deviate significantly from the true distribution, leading to unsatisfactory gate assignment plans. Given that the RF method can handle noise and outliers in the data by combining the prediction results of multiple decision trees, it offers strong robustness and accuracy. Therefore, this study proposes an RF-based ETO method to address this research gap.

Although the ETO approach often provides better decisions, it requires generating a large number of scenarios based on the distributions of uncertain parameters, resulting in poor scalability and computational intractability of the SP models (Yang et al., 2024; Wu et al., 2024). Therefore, when using the ETO approach, adopting an effective scenario selection strategy to sample an appropriate number of scenarios and their associated probabilities is also essential for making satisfactory decisions (Yan

et al., 2022; Yang et al., 2024). The SAA method is used as a scenario selection strategy by Yan et al. (2022) and Yang et al. (2024). The SAA method is based on Monte Carlo simulation, which extracts multiple samples from the uncertainty distribution, generates a set of random scenarios, and uses the average of these scenarios to approximate the objective function of the original problem. Although the SAA method is straightforward, its results may exhibit deviations when the sample size is insufficient. Additionally, increasing the sample size for complex optimisation problems can substantially escalate computational complexity. This study proposes a clustered-based scenario reduction (CSR) method to address these limitations. The K-means method embedded in the CSR method generates more evenly distributed samples, better capturing the distribution characteristics of random variables (Kim et al., 2023b; Wu et al., 2024). Moreover, due to the more uniform distribution of samples, the CSR method requires a smaller sample size while maintaining performance.

The AGAP exhibits an NP-hard nature (Karsu et al., 2021; Li et al., 2021). Given that the proposed SP model for AGAP further considers the uncertainty of aircraft arrival times, achieving an optimal solution can be challenging. Existing commercial mixed-integer programming (MIP) solvers can only handle small-size instances of this model within limited CPU time. Şeker and Noyan (2012) proposed tabu search heuristics to efficiently provide acceptable solutions for the SP model of the AGAP. Nevertheless, in practice, gate assignment is not typically a real-time optimisation problem, as gate plans are usually made hours or even a day before the scheduled time. Therefore, using an exact algorithm that can provide an optimal solution is more advisable (Chen et al., 2024). The studied AGAP consists of three classes of decisions, i.e., aircraft-to-gate assignment decisions, aircraft sequencing decisions, and aircraft scheduling decisions, which have a good decomposition structure. The Benders decomposition (BD) methods are widely used to decompose the SP models into a master problem and several subproblems (Rahmaniani et al., 2018). The master

problem and subproblems are solved iteratively, and in each iteration, Benders cuts are generated and incorporated into the master problem. Solving the master problem of the BD method is time-consuming because it is usually an integer programming problem, and the Benders cuts added in each iteration increase its complexity (Rei et al., 2009). Instead of solving the mixed integer linear programming master problem in each iteration, the BBC method constructs a single branch-and-cut tree, and the Benders cuts are added to the unfathomed nodes in the branch-and-cut tree after finding an integer feasible solution (Gendron et al., 2016). Additionally, due to parts of the original objective function being projected out in the master problem, the quality of lower bound (LB) may be low initially, resulting in a large optimality gap in the initial stages (Rahmaniani, Ragheb and Crainic, Teodor Gabriel and Gendreau, Michel and Rei, Walter, 2017; Rahmaniani et al., 2018). Therefore, a lot of Benders cuts are required to close the gap (Adulyasak et al., 2015). The lower bound lifting (LBL) cuts with some information about the original objective function can be used as initial cuts and added to the master problem to improve the LB, where the LBL cuts represent the LB of arc costs (Adulyasak et al., 2015; Wu et al., 2022). When using the BBC method, we incorporate the LBL cuts at the root node of the branch-and-cut tree. Moreover, the original single Benders optimality cuts are weak. To enhance the strength of original cuts, we employ a multiple-cut strategy that incorporates more or at least the same amount of information into the unfathomed nodes in the branch-and-cut tree (Maheo et al., 2019; Shehadeh, 2023). This approach facilitates the generation of more specific Benders optimality cuts, potentially leading to improved LB and faster convergence rate (Shehadeh, 2023).

Our main contributions are summarised as follows:

(i) We develop two prescriptive analytics approaches for AGAP under aircraft arrival time uncertainty. In the PTO approach, the uncertain arrival time of each aircraft is assumed to be identical to the predicted value provided by ML methods. However,

the accuracy of predictions impacts the performance of the airport gate assignment plans offered by the PTO approach, and inaccurate predictions may result in suboptimal plans. The ETO approach is further utilised for AGAP to deal with this issue, where the first step involves estimating the distribution of uncertain aircraft arrival times, and the second step solves the SP model for AGAP based on the estimated distribution. This study employs KNN and RF methods to predict aircraft arrival times and estimate their distributions. These ML methods are well-constructed and validated. To the best of our knowledge, this is the first time that prescriptive analytics approaches are implemented in the AGAP.

(ii) Although the ETO approach often yields superior decisions, it necessitates generating a substantial number of scenarios based on the distributions of uncertain parameters, resulting in poor scalability and computational intractability of the SP models. To address this issue, we develop a CSR method for scenario selection, which only requires fewer scenarios to enable the ETO approach to achieve good performance in airport gate assignment planning. In addition, we develop a BBC method to efficiently solve the SP model of AGAP, in which the LBL cuts are added to the master problem and the multi-cut strategy is used to generate more specific Benders optimality cuts to improve LB and faster convergence rate.

(iii) We evaluate the performance of the proposed prescriptive analytics approaches and solution method on test instances generated from real-world data from Xiamen Gaoqi International Airport (XMN), an important airport on the southeast coast of China. In particular, we compare the computational performance of the BBC method against the commercial solver. The results demonstrate highly significant statistical improvements in the proposed BBC method compared to the benchmark method. After analysing the effects of ML methods and scenario selection strategies on the ETO approach, we compare the performance of airport gate assignment plans generated by prescriptive analytics approaches with those from other optimisation approaches.

Additionally, we examine the impact of airport gate controller preference levels and provide managerial insights that are practically useful for decision-making in airport gate assignments.

The remainder of this chapter is organised as follows. Section 3.2 describes the problem background and the corresponding mathematical formulations of the AGAP. In Section 3.3, the ML methods are constructed and validated using the accessible data from XMN. Section 3.4 presents the prescriptive analytics approaches for AGAP. The numerical experiments are conducted in Section 3.5, followed by conclusions and future directions in Section 3.6.

## 3.2 Problem setting for AGAP

This section describes the AGAP and presents the corresponding notations and mathematical formulation. Consider a set $I$ that contains arriving aircraft. According to the International Civil Aviation Organisation (ICAO) regulations, aircraft in $I$ can be classified into types A, B, C, D, E, and F based on their wingspan (Bi et al., 2022). Types C, D, and E are the most common aircraft types, with wingspans ranging from 24 to 65 meters. Therefore, our AGAP considers these three most frequent aircraft types and the corresponding contact gates that can accommodate them. However, our AGAP can be easily extended to consider more or all aircraft types if needed. The sets of E-type, D-type, and C-type aircraft are denoted as $I^{\mathrm{E}}$, $I^{\mathrm{D}}$, and $I^{\mathrm{C}}$, respectively. We have $I = I^{\mathrm{E}} \cup I^{\mathrm{D}} \cup I^{\mathrm{C}}$. Besides, the aircraft are either domestic or international. The sets of domestic and international aircraft are separately denoted as $I^{\mathrm{Dom}}$ and $I^{\mathrm{Int}}$, where $I = I^{\mathrm{Dom}} \cup I^{\mathrm{Int}}$. Each aircraft $i \in I$ must be assigned to a gate $g \in G \cup \{r\}$ and occupies its assigned gate for a duration $O_i$, where $G$ denotes the set of contact gates, and $r$ represents the apron. In our AGAP, the compatibility of contact gates is taken into account. The three contact gate sets compatible with E-type, D-type,

and C-type aircraft are denoted as $G^{\mathrm{E}}$, $G^{\mathrm{D}}$ and $G^{\mathrm{C}}$, respectively. E-type aircraft can only be assigned to a contact gate $g \in G^{\mathrm{E}}$, D-type aircraft can be assigned to contact gate $g \in G^{\mathrm{E}} \cup G^{\mathrm{D}}$, and C-type aircraft can be assigned to all contact gates (Dell'Orco et al., 2017; Bi et al., 2022). Furthermore, in airports handling both domestic and international aircraft, the gates for international and domestic aircraft are separated due to different control procedures (Karsu et al., 2021). The contact gate sets are compatible with domestic and international aircraft, which are separately denoted as $G^{\mathrm{Dom}}$ and $G^{\mathrm{Int}}$. The apron $r$ has unlimited capacity and can accommodate any aircraft type (Daş et al., 2020). The contact gates are further assumed to be independent, thus eliminating interference between contact gates (Li et al., 2021). Each contact gate $g \in G$ can be occupied by one aircraft at a time. The notation $x_i^g$ equals to 1 if aircraft $i$ assigned to a gate $g$; otherwise, it equals 0. A buffer time $B$ is required between two aircraft assigned to one contact gate consecutively. Each contact gate begins with a dummy starting aircraft $s$ and finishes with a dummy ending aircraft $e$. The notation $y_{ij}^g$ equals 1 if aircraft $i$ and $j$ are assigned to the same contact gate $g$, with $i$ immediately preceding $j$; otherwise, it equals 0. We define $A_i$ as the estimated arrival time of aircraft $i$. The parking time $t_i$ of aircraft $i$ should be no earlier than $A_i$. The delay time $d_i$ of aircraft $i$ is defined as the difference between related $t_i$ and $A_i$. It should be noted that we do not impose an upper bound (UB) on the park time $t_i$ for each aircraft. This decision accounts for the possibility of an arriving aircraft waiting at its assigned gate for an extended period while the departing aircraft currently occupying the contact gate completes its pushback, although such occurrences are infrequent (Li et al., 2021). The objective function of the AGAP is to minimise the costs of assigning aircraft to the apron $r$, and the costs caused by aircraft delays. The apron $r$, a type of gate without the jet bridge, is typically not preferred by airlines for aircraft assignments under normal circumstances because it increases passenger boarding and disembarking time (Daş et al., 2020; Karsu et al., 2021). Therefore, for aircraft $i$ allocated to the

Table 3.1: Notations and definitions for AGAP.

| Notation | Definition |
|---|---|
| **Sets** | |
| $I^{\mathrm{E}}$ | The set of E-type aircraft. |
| $I^{\mathrm{D}}$ | The set of D-type aircraft. |
| $I^{\mathrm{C}}$ | The set of C-type aircraft. |
| $I^{\mathrm{Dom}}$ | The set of domestic aircraft. |
| $I^{\mathrm{Int}}$ | The set of international aircraft. |
| $I$ | The set of all aircraft. |
| $G^{\mathrm{E}}$ | The set of gates compatible with E-type, D-type, and C-type aircraft. |
| $G^{\mathrm{D}}$ | The set of gates compatible with D-type and C-type aircraft. |
| $G^{\mathrm{C}}$ | The set of gates compatible with C-type aircraft. |
| $G^{\mathrm{Dom}}$ | The set of gates compatible with domestic aircraft. |
| $G^{\mathrm{Int}}$ | The set of gates compatible with international aircraft. |
| $G$ | The set of all contact gates. |
| **Parameters** | |
| $A_i$ | Estimated arrival time of aircraft $i$. |
| $O_i$ | Duration of aircraft $i$ occupying a gate. |
| $B$ | Buffer time between two consecutive aircraft assigned to one contact gate $g$. |
| $M$ | A sufficiently large number. |
| $C_i^{\mathrm{apron}}$ | Cost of assigning aircraft $i$ to the apron. |
| $C_i^{\mathrm{delay}}$ | Unit time delay cost for aircraft $i$. |
| **Decision variables** | |
| $x_i^g$ | 1, if aircraft $i$ assigned to a gate $g$; 0, otherwise. |
| $y_{ij}^g$ | 1, if aircraft $i$ and $j$ assigned to one contact gate $g$, and $i$ immediately precedes $j$; 0, otherwise. |
| $t_i$ | Park time of aircraft $i$. |
| $d_i$ | Delay time of aircraft $i$. |

apron, a penalty cost $C_i^{\mathrm{apron}}$ is incurred. Additionally, $C_i^{\mathrm{delay}}$ represents the unit time delay cost for aircraft $i$. For clarity, we provide the notations of sets, parameters, and decision variables used in this study in Table 3.1.

Based on the problem description and notations provided above, the mathematical

formulations of the AGAP are written as follows:

$$\min \quad \sum_{i \in I} C_i^{\text{apron}} x_i^r + \sum_{i \in I} C_i^{\text{delay}} d_i \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{g \in G \cup \{r\}} x_i^g = 1, \quad \forall i \in I, \tag{3.1b}$$

$$x_i^g \leq 0, \quad \forall g \in G^{\text{D}} \cup G^{\text{C}}, \forall i \in I^{\text{E}}, \tag{3.1c}$$

$$x_i^g \leq 0, \quad \forall g \in G^{\text{C}}, \ \forall i \in I^{\text{D}}, \tag{3.1d}$$

$$x_i^g \leq 0, \quad \forall g \in G^{\text{Int}}, \ \forall i \in I^{\text{Dom}}, \tag{3.1e}$$

$$x_i^g \leq 0, \quad \forall g \in G^{\text{Dom}}, \ \forall i \in I^{\text{Int}}, \tag{3.1f}$$

$$x_i^g = \sum_{j \in I \cup \{e\} \setminus \{i\}} y_{ij}^g, \quad \forall g \in G, \ \forall i \in I, \tag{3.1g}$$

$$\sum_{j \in I \cup \{e\}} y_{sj}^g = 1, \quad \forall g \in G, \tag{3.1h}$$

$$\sum_{i \in I \cup \{s\}} y_{ie}^g = 1, \quad \forall g \in G, \tag{3.1i}$$

$$\sum_{j \in I \cup \{s\} \setminus \{i\}} y_{ji}^g = \sum_{j \in I \cup \{e\} \setminus \{i\}} y_{ij}^g, \quad \forall g \in G, \ \forall i \in I, \tag{3.1j}$$

$$t_i \geq A_i, \quad \forall i \in I, \tag{3.1k}$$

$$d_i \geq t_i - A_i, \quad \forall i \in I, \tag{3.1l}$$

$$t_i + O_i + B - t_j \leq M(1 - y_{ij}^g), \quad \forall g \in G, \ \forall i \in I, \ \forall j \in I, \ i \neq j, \tag{3.1m}$$

$$x_i^g \in \{0,1\}, \quad \forall g \in G \cup \{r\},\ \forall i \in I, \tag{3.1n}$$

$$y_{ij}^g \in \{0,1\}, \quad \forall g \in G,\ \forall i \in I \cup \{s\},\ \forall j \in I \cup \{e\},\ i \neq j, \tag{3.1o}$$

$$t_i \in \mathbb{R}^+, \quad \forall i \in I, \tag{3.1p}$$

$$d_i \in \mathbb{R}^+, \quad \forall i \in I. \tag{3.1q}$$

The objective function (3.1a) minimises the costs of assigning aircraft to the apron and the aircraft delay costs. Constraints (3.1b) ensure that aircraft should be assigned to contact gates or the apron. Constraints (3.1c) to (3.1f) ensure that the aircraft can only be assigned to the compatible gates. Constraints (3.1g) assign aircraft $i$ to contact gate $g$, which relates the two decision variables $x_i^g$ and $y_{ij}^g$. Constraints (3.1h) and (3.1i) ensure the first and last aircraft served by each contact gate are dummy starting aircraft $s$ and dummy ending aircraft $e$, respectively. Constraints (3.1j) ensure contact gate flow conservation. Constraints (3.1k) ensure the park time $t_i$ of aircraft i should be larger than its arrival time $A_i$. Constraints (3.1l) evaluate the delay time for each aircraft. Constraints (3.1m) guarantee the time consistency. Constraints (3.1n) to (3.1q) determine the domain of decision variables.

We then discuss the selection of the $M$ value in Constraints (3.1m), which is related to the maximum of the term $(t_i + O_i + B)$. Specifically, we aim to find an UB for $\max_{i\in I}\{t_i + O_i\} + B$. Among these, the term $\max_{i\in I}\{t_i + O_i\}$ can be regarded as the pushback time of the latest pushback aircraft. We consider a worst case in which all aircraft are assigned to contact gates, and the latest arrival is served first. In this case, we can determine the UB of the latest aircraft pushback time as $\max_{i\in I}\{T_i\} + \sum_{i\in I} O_i + (|I| - 1)\, B$, we thus have $\max_{i\in I}\{t_i + O_i\} + B \leq \max_{i\in I}\{T_i\} + \sum_{i\in I} O_i + |I|B$. Therefore, selecting $M = \max_{i\in I}\{T_i\} + \sum_{i\in I} O_i + |I|B$ is sufficient for Constraints (3.1m) in the deterministic model (3.1).

## 3.3 ML methods for the aircraft arrival time prediction

In the mathematical formulations for AGAP provided in Section 3.2, airport gate assignment plans are made based on the estimated arrival time $A_i$ for each aircraft. However, the estimated arrival time of aircraft is often uncertain in actual operations (Şeker and Noyan, 2012; Ng et al., 2017; Khassiba et al., 2020, 2022). The arrival time of each aircraft can be modelled as a random variable $\widetilde{A}_i$ to address the uncertainty. When airport gate controllers can access additional information beyond historical data on aircraft arrival times, ML methods can be used to predict parameter values or estimate their distributions more accurately. With the support of these predictions, optimisation approaches can provide airport gate assignment plans closer to the actual scenarios on the day of operation. The predicted arrival time $\hat{A}_i$ of aircraft $i$ can be obtained by adding the predicted arrival time deviation value $\hat{a}_i$ to the estimated arrival time $A_i$, which is represented as $\hat{A}_i = A_i + \hat{a}_i$.

The experimental data and several selected features are presented in Subsection 3.3.1. To develop ML methods with good predictive performance, we fine-tune the hyperparameters of the ML methods in Subsection 3.3.2. Finally, we evaluate the performance of these ML methods for aircraft arrival time in Subsection 3.3.3.

### 3.3.1 Dataset and feature engineering

The dataset used in this study was collected from XMN between September 1st and October 31st, 2023. It contains 15,031 records of arriving aircraft, each of which contains relevant aircraft information. Given the significant role of aviation meteorological data in predicting aircraft arrival times (Deng et al., 2024), we further incorporate terminal area forecast (TAF) data into the training dataset. The TAF data include details such

as maximum and minimum temperatures, humidity, air pressure, wind direction, wind speed, and other relevant parameters during the forecast period.

To construct the ML methods, we choose 15 features from the original dataset. These features are presented in Table 3.2 along with their data type, encoding method, and statistical information. We identify missing values that exist in features such as "Aircraft type", "Route distance", and "Fuel load". To address these, we fill in the missing values using the median value for "Aircraft type" and the mean value for both "Route distance" and "Fuel load". Notably, the "Aircraft type", "Domestic/International", "Weather", and "Wind direction" features are expressed in a literal format, necessitating their conversion to numerical data. In the context of aircraft type, the arriving aircraft at XMN primarily fall into three categories: E-type, D-type, and C-type. We then encode these types as 3, 2, and 1. The weather includes sunny, rainy, cloudy, foggy. Additionally, 16 wind directions are considered. For weather and wind directions, we adopt one-hot encoding. We designate the data collected between September 1st and October 30th, 2023, as the dataset $D$ with 14,783 records of arriving aircraft. We randomly split 80% of the data in $D$ as the training set $D^{\text{Training}}$ for training the ML models, while the remaining 20% as the testing set $D^{\text{Testing}}$ to evaluate their performance. Additionally, the data from October 31st, 2023, is used to generate the test instances.

### 3.3.2 Hyperparameters tuning

Within the KNN method, the only hyperparameter is $K$ (i.e., $n\_neighbours$), which represents the number of neighbours the KNN method takes into account. Opting for a very small $K$ value may render the model overly sensitive to noise while selecting a larger $K$ value reduces noise impact and increases model complexity. Consequently, careful tuning of the $K$ value is essential in practical applications. In the RF method, more hyperparameters need to be considered. We list some crucial hyperparameters of

Table 3.2: Features in ML methods for AGAP.

| Feature name | Data type | Encoding | Null count |
|---|---|---|---|
| Aircraft type | Object | Label encoding | 32 |
| Domestic/International | Object | One-hot encoding | 0 |
| Estimated take-off time | Numerical | | 0 |
| Estimated landing time | Numerical | | 0 |
| Straight line distance | Numerical | | 0 |
| Route distance | Numerical | | 886 |
| Estimated flight time | Numerical | | 0 |
| Fuel load | Numerical | | 892 |
| Weather | Object | One-hot encoding | 0 |
| High temperature | Numerical | | 0 |
| Low temperature | Numerical | | 0 |
| Humidity | Numerical | | 0 |
| Barometer | Numerical | | 0 |
| Wind direction | Object | One-hot encoding | 0 |
| Wind speed | Numerical | | 0 |

the RF method in the following. First is the number of decision trees in the forest (denoted by $n\_estimators$). Choosing an appropriate value for $n\_estimators$ can balance underfitting and overfitting. Generally, larger values of $n\_estimators$ improve the performance of the RF method but also increase computational cost. Second, a leaf node should contain the minimum number of samples (denoted by $min\_samples\_leaf$). A leaf node can be further split if and only if it contains at least $min\_samples\_leaf$ data. Third, the minimum number of samples a node should contain (denoted by $min\_samples\_split$). A node can be further split if and only if it includes at least $min\_samples\_split$ data. Finally, the maximum depth of decision trees (denoted by $max\_depth$). Limiting the maximum depth can prevent overfitting when there are many samples or features.

The hyperparameters tuned for the KNN and RF methods are detailed in Table 3.3. These best hyperparameter values are determined using a grid search method with 5-fold cross-validation on the training dataset $D^{\text{Training}}$.

Table 3.3: Best hyperparameter values of the ML methods for AGAP.

| Hyperparameters | Search space | Best value |
|---|---|---|
| **KNN method** | | |
| *n_neighbours* | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] | 8 |
| **RF method** | | |
| *n_estimators* | [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000] | 700 |
| *min_samples_split* | [2, 3, 4, 5, 6, 7, 8, 9, 10] | 9 |
| *min_samples_leaf* | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] | 2 |
| *max_depth* | [10, 20, 30, 40, 50, 60, 70, 80, 90, 100] | 50 |

### 3.3.3 Performance of ML methods

The performance of the ML methods is evaluated using mean absolute error (MAE), mean square error (MSE), and root MSE (RMSE), which are commonly used evaluation metrics when ML methods are used to predict aircraft arrival times (Wang et al., 2020; Deng et al., 2024). We also illustrate the biases of the ML methods using the mean bias error (MBE). Considered a data set with $n$ samples, where $\bar{A}_i$ and $\hat{A}_i$ are the actual and predicted arrival time of aircraft $i \in \{1, \ldots, n\}$, respectively. These metrics are defined as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\bar{A}_i - \hat{A}_i\right|. \tag{3.2}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\bar{A}_i - \hat{A}_i\right)^2. \tag{3.3}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\bar{A}_i - \hat{A}_i\right)^2}. \tag{3.4}$$

$$\text{MBE} = \frac{1}{n}\sum_{i=1}^{n}\left(\bar{A}_i - \hat{A}_i\right). \tag{3.5}$$

After predicting the arrival time deviation value $\hat{a}_i$ using the ML methods, we calculate the predicted arrival time $\hat{A}_i$ using the formula $\hat{A}_i = A_i + \hat{a}_i$, where the estimated arrival time $A_i$ is assumed to be known in advance. We further use the

estimated arrival time $A_i$ to replace the predicted arrival time $\hat{A}_i$ in Equation (3.2) to (3.5) for calculating the MAE, MSE, RMSE, and MBE values as benchmarks. Based on the prediction results presented in Table 3.4, both KNN and RF methods are better than the benchmark, as they yield smaller MAE, MSE, and RMSE values. These show that the predicted arrival times provided by ML methods are more accurate than the estimated arrival times. We also find that the prediction accuracy of the RF method is better than that of the KNN method. The MAE value of the RF method is controlled within 15 minutes. Given that an aircraft is considered "delayed" if its actual arrival time exceeds the estimated arrival time by more than 15 minutes, this demonstrates that the arrival time predictions made by the RF method are relatively accurate from the perspective of the MAE indicator. In addition, the MBE value of the KNN method is -2.73, indicating that its predicted values are, on average, 2.73 units lower than the actual values. This negative deviation reveals that the KNN method generally underestimates the arrival time on this dataset. On the other hand, the MBE value of the RF method is 0.40, suggesting that its predicted values are, on average, 0.40 units higher than the actual values. This positive deviation indicates a slight overestimation by the RF method, though the overall deviation remains small. The KNN method's larger MBE value signifies its poorer performance in predicting aircraft arrival time, potentially due to systematic underestimation. Conversely, the RF model exhibits a smaller deviation, highlighting its more stable performance and closer alignment with the actual situation.

After training, testing and validating the ML methods, we can utilise them to predict values or estimate distributions for uncertain parameters more accurately. These predicted values or estimated distributions can then be used as inputs for prescriptive analytics approaches provided in Section 3.4, enabling the development of airport gate assignment plans that are more closely aligned with the actual conditions on the operational day.

Table 3.4: Prediction results on the testing dataset for AGAP.

| Metrics | Benchmark | KNN | | RF | |
|---|---|---|---|---|---|
| | Value | Value | Reduction (%) | Value | Reduction (%) |
| MAE | 17.65 | 15.73 | 10.88 | 14.81 | 16.10 |
| MSE | 1129.17 | 961.32 | 14.87 | 822.65 | 27.15 |
| RMSE | 33.60 | 31.01 | 7.73 | 28.68 | 14.65 |
| MBE | 15.40 | -2.73 | - | 0.41 | - |

## 3.4 Prescriptive analytics approaches for AGAP

This section proposes two prescriptive analytics approaches for AGAP based on the ML methods introduced in Section 3.3. The PTO approach for AGAP is presented in Subsection 3.4.1. And the ETO approach for AGAP is provided in Subsection 3.4.2.

### 3.4.1 PTO approach for AGAP

Using the predicted arrival time $\hat{A}_i$ provided by ML methods, the PTO approach can be applied, where the random variable $\widetilde{A}_i$ is assumed to be identical to the predicted value. In this case, the distribution of $\widetilde{A}_i$ can be mathematically presented as $\Pr\left(\widetilde{A}_i = \hat{A}_i\right) = 1$, where $\Pr(\cdot)$ represents the probability an event occurring. By substituting the estimated arrival time $A_i$ in Model (3.1) with the predicted arrival time $\hat{A}_i$, the mathematical formulations of the PTO approach for AGAP is formulated. The framework of the PTO approach driven by RF method for AGAP is provided in Figure 3.1.

### 3.4.2 ETO approach for AGAP

When using the PTO approach for AGAP, each random variable $\widetilde{A}_i$ is assumed to be identical to the predicted arrival time $\hat{A}_i$ provided by ML methods. Consequently, the accuracy of these prediction results impacts the performance of the airport gate as-

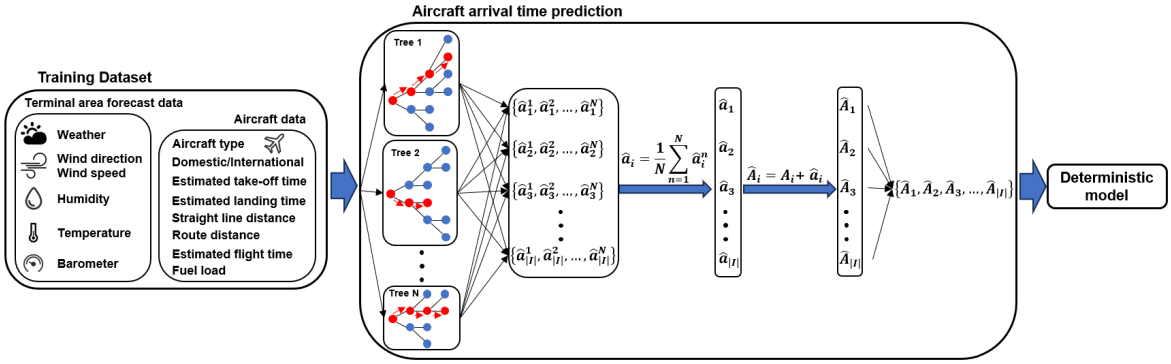


Figure 3.1: Framework of PTO approach for AGAP driven by RF method.

signment plans, and inaccurate predictions may result in suboptimal plans. The ETO approach can be utilised for AGAP to address these issues. The framework of this approach, driven by the RF method, is provided in Figure 3.2. In the ETO approach, the first step involves estimating the distribution of unknown parameters $\widetilde{A}_i$, and the second step solves the SP model for AGAP based on the estimated distribution. In the ML methods used in this study, the KNN method computes the prediction result as the mean of the possible arrival times from a predetermined number of neighbours. Meanwhile, the RF method calculates the prediction result by averaging the predicted arrival times from a predetermined number of decision trees. Set $N$ as the predetermined number of neighbours or decision trees. That is, $N$ possible values of the arrival time of aircraft $i$ provided by the ML methods. We then denote $\mathcal{P}(f_i)$ as the set containing $N$ possible arrival times when using input feature vector $f_i$. The vector of the arrival time for all aircraft in $I$ can be written as $\widetilde{A} = \left(\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_{|I|}\right)$. And the Cartesian product $\Phi\left(\widetilde{A}\right) = \left\{\mathcal{P}(f_1) \times \mathcal{P}(f_2) \times \cdots \times \mathcal{P}\left(f_{|I|}\right)\right\}$ can be used to approximate the distribution of $\widetilde{A}$ (Yan et al., 2022; Yang et al., 2024). The Cartesian product contains $N^{|I|}$ elements, which is an exponential function of $|I|$ (Yan et al., 2022). Based on this, the uncertainty of aircraft arrival times is represented by a finite set $\Xi = \left\{1,\ 2,\ \ldots,\ \left|\Phi\left(\widetilde{A}\right)\right|\right\}$ of scenarios, where the size of this set is also $N^{|I|}$. We further assume that the scenarios in the set $\Xi$ are independent and identically distributed

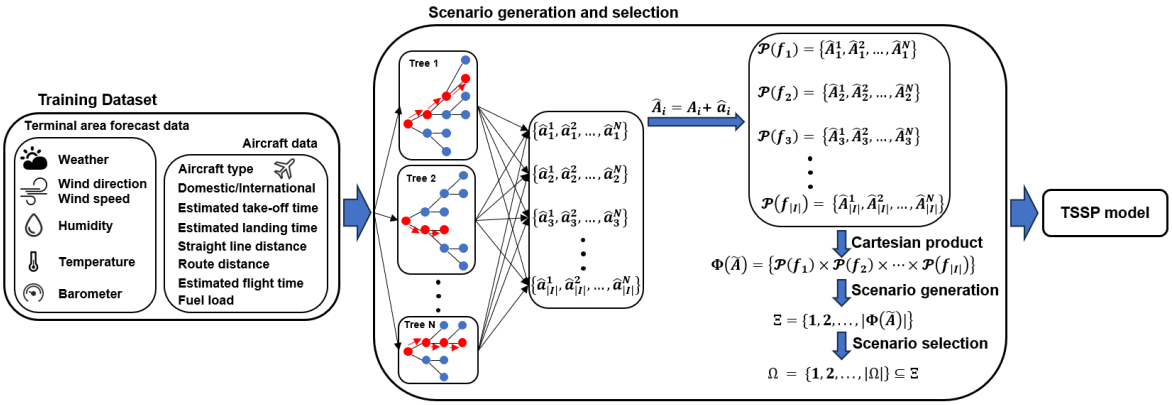


Figure 3.2: Framework of ETO approach for AGAP driven by RF method.

(i.i.d.). In the remainder of this subsection, we first introduce the SP model for AGAP. Then, scenario selection strategies are employed to approximate the original SP model using a set of sample scenarios to enhance computational tractability. Finally, an exact BBC method is developed for the approximated SP model to enhance scalability.

#### 3.4.2.1 SP model for AGAP

We formulate AGAP as an SP model with a two-stage decision-making process. In the first stage, aircraft are assigned to appropriate gates, and aircraft sequencing decisions are made for each gate. In the second stage, aircraft scheduling decisions are determined after the arrival time $A_i^\xi$ of aircraft i is realised under $\xi \in \Xi$. In the SP model, the park time $t_i^\xi$ of aircraft $i$ under scenario $\xi$ should be no earlier than its arrival time $A_i^\xi$. The delay time $d_i^\xi$ of aircraft $i$ under scenario $\xi$ is defined as the difference between its park time $t_i^\xi$ and its arrival time $A_i^\xi$. The mathematical formulations of the SP model are written as follows:

$$\min \quad \sum_{i \in I} C_i^{\text{apron}} x_i^r + \mathbb{E}_\xi \left[ Q(\mathbf{y}, \xi) \right] \tag{3.6a}$$

$$\text{s.t.} \quad \text{Constraints (3.1b)-(3.1j), (3.1n), (3.1o),} \tag{3.6b}$$

where the term $\mathbb{E}_\xi[Q(\mathbf{y},\xi)]$ represents the expected value of the second-stage objective function $Q(\mathbf{y},\xi)$. For a feasible first-stage solution $\mathbf{y}$ and the realisation of scenario $\xi$, $Q(\mathbf{y},\xi)$ is formulated as follows:

$$Q(\mathbf{y},\xi) = \min \sum_{i\in I} C_i^{\text{delay}} d_i^\xi \tag{3.7a}$$

$$\text{s.t.} \quad t_i^\xi \geq A_i^\xi, \quad \forall i \in I, \tag{3.7b}$$

$$d_i^\xi \geq t_i^\xi - A_i^\xi, \quad \forall i \in I, \tag{3.7c}$$

$$t_i^\xi + O_i + B - t_j^\xi \leq M(1 - y_{ij}^g), \quad \forall g \in G,\ \forall i \in I,\ \forall j \in I,\ i \neq j, \tag{3.7d}$$

$$t_i^\xi \in \mathbb{R}^+, \quad \forall i \in I, \tag{3.7e}$$

$$d_i^\xi \in \mathbb{R}^+, \quad \forall i \in I. \tag{3.7f}$$

#### 3.4.2.2 Scenario selection strategies

Recall that $N^{|I|}$ scenarios are generated using the Cartesian product of prediction results from the KNN or RF methods, and are subsequently included in the set $\Xi$. Solving all the scenarios within $\Xi$ can render the SP model for AGAP intractable. Therefore, appropriate scenario selection strategies should be employed to generate a sample set $\Omega = \{1, 2, ..., |\Omega|\}$ to approximate the set $\Xi$. In this way, the SP model (3.6) can be reformulated by the scenario selection strategies into Model (3.8), where $p^\omega$ is the scenario probability. The selection of $M$ for Constraints (3.8e) is similar to that for Constraints (3.1m). Since the scenarios are independent, we can initially determine the appropriate $M$ for each scenario $\omega \in \Omega$, represented by $M^\omega$, where $M^\omega = \max_{i\in I}\{T_i^\omega\} + \sum_{i\in I} O_i + |I|B$. And the choice of $M = \max_{\omega\in\Omega}\{M^\omega\}$ is sufficient

for Constraints (3.8e) in the SP model (3.8).

Model (3.8) is a mixed-integer linear programming (MILP) model and can be solved directly by commercial MIP solvers.

$$\min \quad \sum_{i\in I} C_i^{\text{apron}} x_i^r + \sum_{\omega\in\Omega} p^\omega \Big(\sum_{i\in I} C_i^{\text{delay}} d_i^\omega\Big) \tag{3.8a}$$

$$\text{s.t.} \quad \text{Constraints (3.1b)-(3.1j), (3.1n), (3.1o),} \tag{3.8b}$$

$$t_i^\omega \geq A_i^\omega, \quad \forall i \in I,\ \forall \omega \in \Omega, \tag{3.8c}$$

$$d_i^\omega \geq t_i^\omega - A_i^\omega, \quad \forall i \in I,\ \forall \omega \in \Omega, \tag{3.8d}$$

$$t_i^\omega + O_i + B - t_j^\omega \leq M(1 - y_{ij}^g), \quad \forall g \in G,\ \forall i \in I,\ \forall j \in I,\ i \neq j,\ \forall \omega \in \Omega, \tag{3.8e}$$

$$t_i^\omega \in \mathbb{R}^+, \quad \forall i \in I,\ \forall \omega \in \Omega, \tag{3.8f}$$

$$d_i^\omega \in \mathbb{R}^+, \quad \forall i \in I,\ \forall \omega \in \Omega. \tag{3.8g}$$

Previous studies used the SAA method as a scenario selection strategy (Yan et al., 2022; Yang et al., 2024). The SAA method is a widely used approach that randomly samples equiprobable scenarios (Kleywegt et al., 2002). This method generates the sample set $\Omega$ by randomly selecting scenarios from $\Xi$. The scenario probability $p^\omega$ of each scenario $\omega \in \Omega$ is set as $\frac{1}{|\Omega|}$.

In this study, another scenario selection strategy known as the CSR method is also used. The CSR method involves clustering similar scenarios and selecting a representative scenario from each cluster, where the number of cluster centres is set to $|\Omega|$ (Kim et al., 2023b; Wu et al., 2024). It is worth noting that the number of cluster centres equals the scenario number output by the CSR method. We utilise the

K-means method to cluster the scenarios in the set $\Xi$ and determine the cluster centres. Subsequently, we employ the Euclidean distance for each cluster centre to select the scenario closest to the centre and include that scenario in the sample set $\Omega$. Let $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_{|\Omega|}\}$ be a set of $|\Omega|$ clusters obtained by the K-means method, the scenario probability of each scenario $\omega \in \Omega$ is $p^\omega = \frac{|\mathcal{C}_\omega|}{|\Xi|}$. However, as previously mentioned, the size of the set $\Xi$ is $N^{|I|}$, an exponential function of $I$, which is usually extremely large. Directly applying the CSR method requires a significant amount of CPU time to complete the clustering process. Consequently, we first employ the SAA method to extract a subset $\Xi^{\text{sub}}$ from the original set $\Xi$, and subsequently apply the CSR method to this subset.

#### 3.4.2.3 Solution method for SP model of the AGAP

The reformulated SP model (3.8) for AGAP provided in Subsection 3.4.2.2 can be directly solved by a commercial MIP solver. Nevertheless, because of the NP-hard nature of the AGAP, the commercial MIP solver can only deal with small-size instances. To improve the solvability of the ETO approach, we thus propose the exact BBC method. The BBC method decomposes the SP model (3.8) into a master problem and several subproblems. A branch-and-cut MIP solver solves the master problem in the BBC method. Since the time-related constraints in the original SP model are omitted from the master problem, an integer first-stage solution with subtours can be found during the branch-and-cut search. Once an integer first-stage solution with subtours is found, the Dantzig-Fulkerson-Johnson (DFJ) subtour elimination cuts will be generated and added to the unfathomed nodes in the branch-and-cut tree. Otherwise, if an integer first-stage solution without subtours is found, this solution is then employed to solve the subproblems. Since the scenarios are independent and the contact gates do not interfere with each other, the subproblem can be disaggregated into $|\Omega| * |G|$ disconnected subproblems. Then, Benders optimality cuts will be generated and added

to the unfathomed nodes in the branch-and-cut tree. The pseudo-code of the BBC method is presented in Algorithm 1.

**Algorithm 1** The BBC method

```
Solve the master problem (3.9) using a branch-and-cut MIP solver.
If an integer first-stage solution ŷ is found then
    If there are no subtours in ŷ then
        Solve subproblems (3.12) with ŷ.
        Add Benders cuts (3.13) to the unfathomed nodes.
    Else
        Add subtour elimination cuts (3.10) to the unfathomed nodes.
    End if
End if
```

The master problem of the BBC method is formulated as follows:

$$\min \quad \sum_{i \in I} C_i^{\text{apron}} x_i^r + \sum_{\omega \in \Omega} p^\omega \left( \sum_{g \in G} \theta^{g\omega} \right) \tag{3.9a}$$

$$\text{s.t.} \quad \text{Constraints (3.1b)-(3.1j)}, \ \text{(3.1n)}, \ \text{(3.1o)}, \tag{3.9b}$$

$$\sum_{g \in G} \theta^{g\omega} \geq \sum_{g \in G} \sum_{i \in I} \sum_{j \in I \setminus \{i\}} \mathcal{D}_{ij}^\omega y_{ij}^g, \quad \forall \omega \in \Omega, \tag{3.9c}$$

$$\theta^{g\omega} \in \mathbb{R}^+, \quad \forall g \in G, \ \forall \omega \in \Omega, \tag{3.9d}$$

where Constraints (3.9c) are the LBL cuts. Due to parts of the original objective function being projected out in the master problem of the BBC method, the initial quality of the LB may be low, resulting in a large optimality gap in the early stages (Rahmaniani, Ragheb and Crainic, Teodor Gabriel and Gendreau, Michel and Rei, Walter, 2017; Rahmaniani et al., 2018). The LBL cuts that incorporate some information about the original objective function can be added to the master problem to improve the LB of the BBC method (Adulyasak et al., 2015; Wu et al., 2022). Specif-

ically, these LBL cuts represent an LB on the arc costs (Adulyasak et al., 2015). For aircraft $i \in I$ and aircraft $j \in I \setminus \{i\}$ that are sequentially severed by any contact gate $g$, a LB of delay time incurred by this arc $(i, j)$ under scenario $\omega$ is computed as $\mathcal{D}_{ij}^{\omega} = \max\left\{0, C_j^{\text{delay}}\left(A_i^{\omega} + O_i + B - A_j^{\omega}\right)\right\}$. Constraints (3.9d) define the domain of decision variables.

The master problem in the BBC method is solved using a branch-and-cut MIP solver. Once a feasible solution is found in the branch-and-cut tree, the determined aircraft assignment decisions and sequencing decisions are fixed as $\hat{x}_i^g$ and $\hat{y}_{ij}^g$, respectively. Since the time-related constraints in the original SP model are omitted from the master problem, an integer first-stage solution with subtours may be found during the branch-and-cut search. Once such a solution is identified, we start checking the subtours in the solution, and put the found subtours $\int$ into the subtour set $\mathcal{S}$. The DFJ subtour elimination cuts (3.10) are generated based on the subtour set $\mathcal{S}$ and added to the unfathomed nodes in the branch-and-cut tree.

$$\sum_{(i,j)\in\int} y_{ij}^g \leq |\int| - 1, \quad \forall g \in G,\ \forall \int \in \mathcal{S}. \tag{3.10}$$

On the other hand, if an integer first-stage solution without subtours is found, we then solve the subproblem with this solution. Since the scenarios are independent and the contact gates do not interfere with each other, the subproblem can be disaggregated into $|\Omega| * |G|$ disconnected subproblems using a multiple-cut strategy. In the multiple-cut strategy, the unfathomed nodes in the branch-and-cut tree incorporate more or at least the same amount of information, which may lead to an improvement in the LB and a faster convergence rate (Maheo et al., 2019; Shehadeh, 2023). Consider the notation $\hat{\mathcal{H}}^g$, for $g \in G$, which refers to the clusters of aircraft assigned to each contact gate $g$ by the feasible first-stage solution without subtours found during the branch-and-cut search, i.e., $\hat{\mathcal{H}}^g = \{i | \hat{x}_i^g = 1, \forall i \in I\}$. When $\hat{\mathcal{H}}^g \neq \emptyset$, the primal subproblem

for contact gate $g$ under scenario $\omega$ is formulated as follows:

$$\min \quad \sum_{i \in \hat{\mathcal{H}}^g} C_i^{\text{delay}} d_i^\omega \tag{3.11a}$$

$$\text{s.t.} \quad t_i^\omega \geq A_i^\omega, \quad \forall i \in \hat{\mathcal{H}}^g, \tag{3.11b}$$

$$d_i^\omega \geq t_i^\omega - A_i^\omega, \quad \forall i \in \hat{\mathcal{H}}^g, \tag{3.11c}$$

$$t_i^\omega + O_i + B - t_j^\omega \leq M(1 - \hat{y}_{ij}^g), \quad \forall i \in \hat{\mathcal{H}}^g,\ \forall j \in \hat{\mathcal{H}}^g,\ i \neq j, \tag{3.11d}$$

$$t_i^\omega \in \mathbb{R}^+, \quad \forall i \in \hat{\mathcal{H}}^g, \tag{3.11e}$$

$$d_i^\omega \in \mathbb{R}^+, \quad \forall i \in \hat{\mathcal{H}}^g. \tag{3.11f}$$

By introducing dual variables $\pi_i^{g\omega}$, $\sigma_i^{g\omega}$, and $\tau_{ij}^{g\omega}$ for Constraints (3.11b) to (3.11d), the dual subproblem for contact gate $g$ under scenario $\omega$ is written as follows:

$$\min \quad \sum_{i \in \hat{\mathcal{H}}^g} A_i^\omega \pi_i^{g\omega} + \sum_{i \in \hat{\mathcal{H}}^g} A_i^\omega \sigma_i^{g\omega} + \sum_{i \in \hat{\mathcal{H}}^g} \sum_{j \in \hat{\mathcal{H}}^g \setminus \{i\}} \left(M\left(1 - \hat{y}_{ij}^g\right) - O_i - B\right) \tau_{ij}^{g\omega} \tag{3.12a}$$

$$\text{s.t.} \quad \pi_i^{g\omega} + \sigma_i^{g\omega} + \sum_{j \in \hat{\mathcal{H}}^g \setminus \{i\}} \tau_{ij}^{g\omega} - \sum_{j \in \hat{\mathcal{H}}^g \setminus \{i\}} \tau_{ji}^{g\omega} \leq 0, \quad \forall i \in \hat{\mathcal{H}}^g, \tag{3.12b}$$

$$-\sigma_i^{g\omega} \leq C_i^{\text{delay}}, \quad \forall i \in \hat{\mathcal{H}}^g, \tag{3.12c}$$

$$\pi_i^{g\omega} \in \mathbb{R}^+, \quad \forall i \in \hat{\mathcal{H}}^g, \tag{3.12d}$$

$$\sigma_i^{g\omega} \in \mathbb{R}^-, \quad \forall i \in \hat{\mathcal{H}}^g, \tag{3.12e}$$

$$\tau_{ij}^{g\omega} \in \mathbb{R}^-, \quad \forall i \in \hat{\mathcal{H}}^g,\ \forall i \in \hat{\mathcal{H}}^g,\ i \neq j. \tag{3.12f}$$

Recall that we do not impose a UB on the park time $t_i^\omega$ for each aircraft under each scenario, based on the assumption provided by Li et al. (2021). Consequently, the SP model for AGAP has relatively complete recourse property, as all first-stage solutions without subtours are feasible for the second-stage problem. Therefore, in the BBC method, all subproblems are feasible for the first-stage solutions without subtours, and only Benders optimality cuts will be generated during the branch-and-cut search. The Benders optimality cuts are presented as Constraints (3.13), which will be generated and added to the unfathomed nodes in the branch-and-cut tree.

$$\theta^{g\omega} \geq \left( \sum_{i \in I} A_i^\omega \hat{\pi}_i^{g\omega} + \sum_{i \in I} A_i^\omega \hat{\sigma}_i^{g\omega} + \sum_{i \in I} \sum_{j \in I \setminus \{i\}} (M - O_i - B) \hat{\tau}_{ij}^{g\omega} - M \sum_{i \in I} \sum_{j \in I \setminus \{i\}} \hat{\tau}_{ij}^{g\omega} y_{ij}^g \right), \quad \forall g \in G, \ \forall \omega \in \Omega. \tag{3.13}$$

## 3.5 Numerical experiments for AGAP

In this section, we first provide the experimental design. Then, the numerical experiments are conducted to evaluate the performance of the solution methods, ML methods, and scenario selection strategies for the ETO approach of AGAP. Afterwards, we compare the prescriptive analytics approaches with other optimisation approaches for AGAP. We also analyse the impact of subset size in the CSR method, the influence of the decision horizon on gate assignment plan performance, and the effect of gate controller preference levels. Lastly, valuable managerial implications and insights are provided for airport gate controllers.

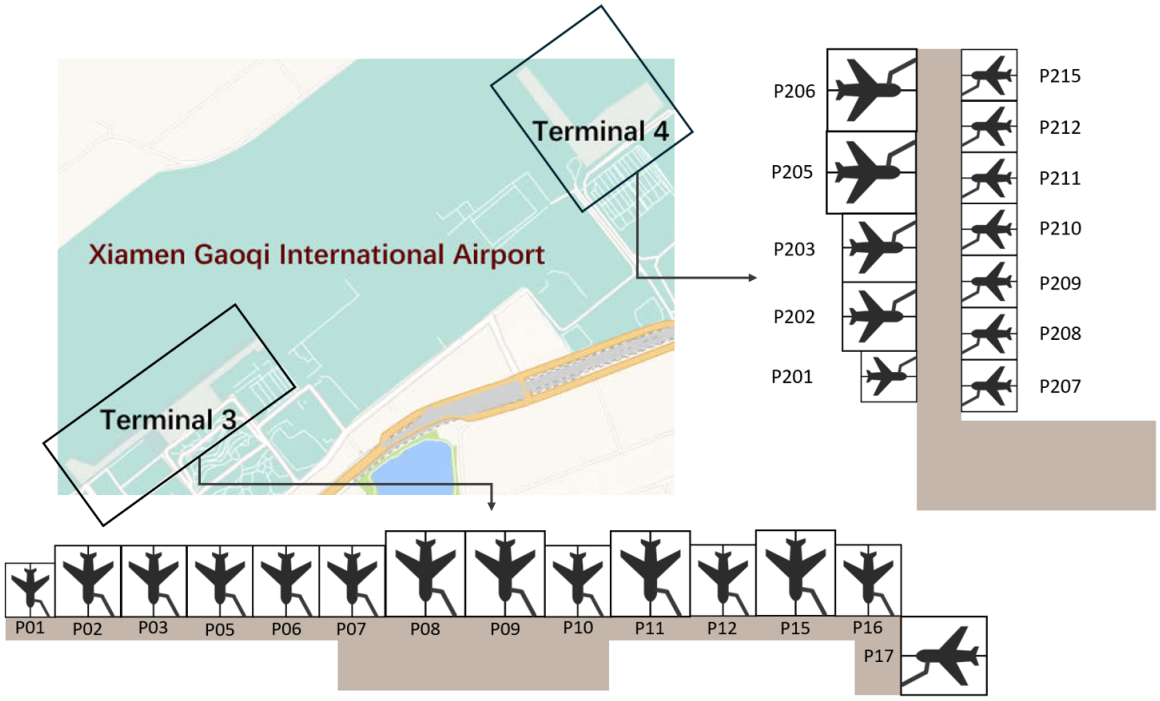


Figure 3.3: Map of Terminals 3 and 4, Xiamen Gaoqi International Airport, China.

### 3.5.1 Experimental design for AGAP

This subsection designs computational experiments to investigate the computational performance of the BBC method and the efficiency of the prescriptive analytics approach. The experiments are based on XMN. As shown in Figure 3.3, XMN consists of two terminals: T3 and T4. Table 3.5 provides details on contact gates.

The test instances are generated based on the realistic data from XMN, on October 31st, 2023. The number of arriving aircraft at hourly intervals is presented in Figure 3.4. Considering real-life gate assignment operations, the airport gate assignment plan, which includes assignment and sequencing decisions, is typically decided by airport gate controllers several hours in advance. We thus use 2-hour, 3-hour, and 4-hour decision horizons to divide the data of this day. After removing duplicate instances and instances with a number of aircraft less than or equal to 1, the total number of

Table 3.5: Detailed information of contact gates at XMN, China.

| Contact gate | Terminal | Domestic/International | Accommodated aircraft type |
|---|---|---|---|
| P01 | 3 | Domestic | C |
| P02 | 3 | Domestic | D |
| P03 | 3 | Domestic | D |
| P05 | 3 | Domestic | D |
| P06 | 3 | Domestic | D |
| P07 | 3 | Domestic | D |
| P08 | 3 | Domestic | E |
| P09 | 3 | Domestic | E |
| P10 | 3 | International | D |
| P11 | 3 | International | E |
| P12 | 3 | International | D |
| P15 | 3 | International | E |
| P16 | 3 | International | D |
| P17 | 3 | International | E |
| P201 | 4 | Domestic | C |
| P202 | 4 | Domestic | D |
| P203 | 4 | Domestic | D |
| P205 | 4 | Domestic | E |
| P206 | 4 | Domestic | E |
| P207 | 4 | Domestic | C |
| P208 | 4 | Domestic | C |
| P209 | 4 | Domestic | C |
| P210 | 4 | Domestic | C |
| P211 | 4 | Domestic | C |
| P212 | 4 | Domestic | C |
| P215 | 4 | Domestic | C |

test instances is 21. Details of each test instance are provided in Table 3.6. We employ the SAA and CSR methods as scenario selection strategies for each test instance. The number of scenarios sampled from $\Xi$ is set to $|\Omega| \in \{5, 10, 25, 50, 75, 100, 200, 300\}$, and the size of the subset used in the CSR method is set to $10^5$.

To consider that delays of larger aircraft with a higher number of passengers are more important (Pohl et al., 2021), we use delay cost coefficients $C_i^{\text{delay}}$ of 1, 2, 3 monetary units for C-type, D-type, and E-type aircraft, respectively. Recognising

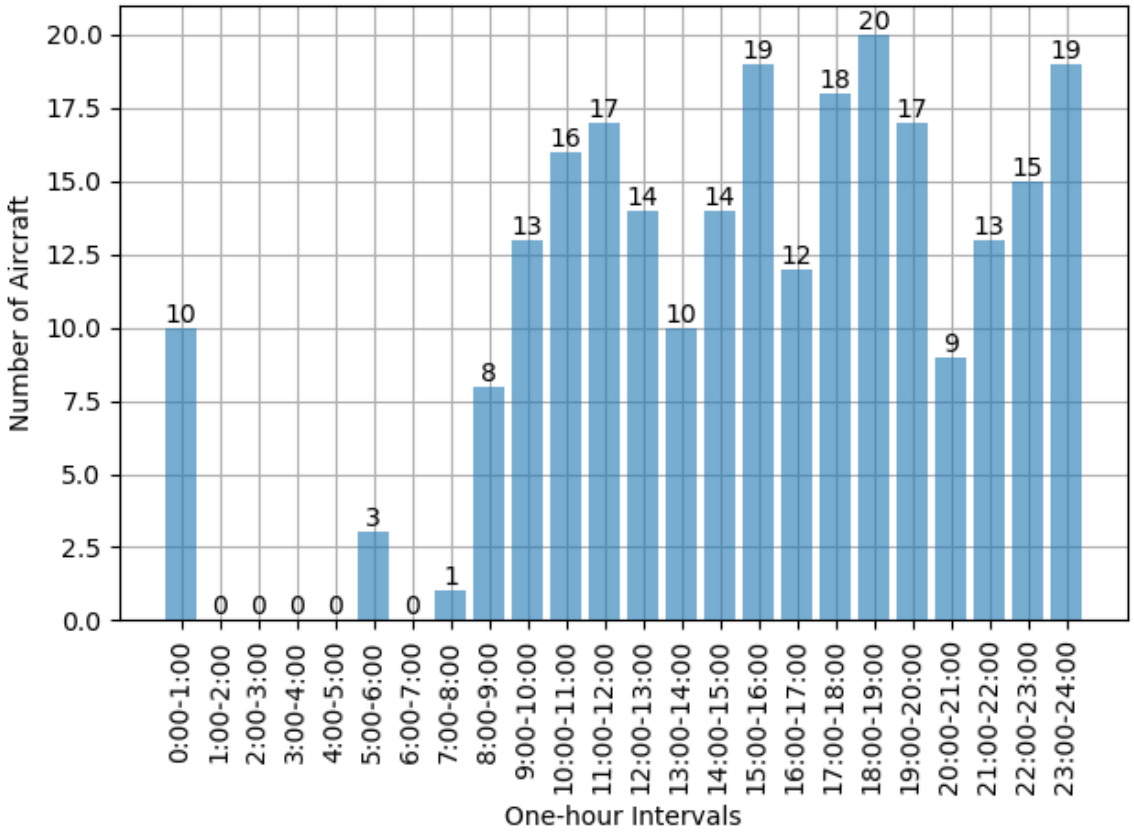


Figure 3.4: Number of arriving aircraft at hourly intervals on 31st October 2023 at XMN, China.

that the apron is inconvenient for both passengers and airlines, penalty costs are often considered to discourage aircraft assignments to the apron. In this study, we set $C_i^{\text{apron}} = \lambda C_i^{\text{delay}}$, where $\lambda \in \{50, 75, 100, 125, 150\}$. As $\lambda$ value increases, the penalty cost associated with assigning an aircraft to the apron also increases. This trend suggests that the airport gate controller prefers not to assign aircraft to the apron. We will discuss the impact of the gate controller preference level in Subsection 3.5.7. The buffer time $B$ is set to 10 minutes (Zhang and Klabjan, 2017).

When using the ETO approach for AGAP, various ML methods and scenario selection strategies are used for generating the sample set $\Omega$. For convenience, we use

Table 3.6: Detailed information of test instances for AGAP.

| Instances | Horizon | Total | Domestic | | | International | | |
|---|---|---|---|---|---|---|---|---|
| | | | E-type | D-type | C-type | E-type | D-type | C-type |
| 1 | 0:00-2:00 | 10 | 0 | 2 | 7 | 1 | 0 | 0 |
| 2 | 4:00-6:00 | 3 | 0 | 0 | 0 | 2 | 0 | 1 |
| 3 | 8:00-10:00 | 21 | 0 | 8 | 13 | 0 | 0 | 0 |
| 4 | 10:00-12:00 | 33 | 2 | 9 | 19 | 1 | 0 | 2 |
| 5 | 12:00-14:00 | 24 | 1 | 8 | 12 | 1 | 1 | 1 |
| 6 | 14:00-16:00 | 33 | 1 | 5 | 24 | 0 | 1 | 2 |
| 7 | 16:00-18:00 | 30 | 1 | 7 | 20 | 0 | 1 | 1 |
| 8 | 18:00-20:00 | 37 | 1 | 13 | 19 | 2 | 1 | 1 |
| 9 | 20:00-22:00 | 22 | 0 | 11 | 7 | 1 | 0 | 3 |
| 10 | 22:00-24:00 | 34 | 0 | 9 | 21 | 1 | 0 | 3 |
| 11 | 6:00-9:00 | 9 | 0 | 4 | 4 | 1 | 0 | 0 |
| 12 | 9:00-12:00 | 46 | 2 | 13 | 28 | 1 | 0 | 2 |
| 13 | 12:00-15:00 | 38 | 2 | 13 | 18 | 1 | 2 | 2 |
| 14 | 15:00-18:00 | 49 | 1 | 7 | 38 | 0 | 1 | 2 |
| 15 | 18:00-21:00 | 46 | 1 | 17 | 22 | 3 | 1 | 2 |
| 16 | 21:00-24:00 | 47 | 0 | 16 | 25 | 1 | 0 | 5 |
| 17 | 4:00-8:00 | 4 | 0 | 0 | 0 | 3 | 0 | 1 |
| 18 | 8:00-12:00 | 54 | 2 | 17 | 32 | 1 | 0 | 2 |
| 19 | 12:00-16:00 | 57 | 2 | 13 | 36 | 1 | 2 | 3 |
| 20 | 16:00-20:00 | 67 | 2 | 20 | 39 | 2 | 2 | 2 |
| 21 | 20:00-24:00 | 56 | 0 | 20 | 28 | 2 | 0 | 6 |

the following notations to represent the combination of these methods: (i) KNN-SAA: using the KNN method for predicting and the SAA method for scenario selection. (ii) KNN-CSR: using the KNN method for predicting and the CSR method for scenario selection. (iii) RF-SAA: using the RF method for predicting and the SAA method for scenario selection. (iv) RF-CSR: using the RF method for predicting and the CSR method for scenario selection.

When using prescriptive analysis approaches, airport gate assignment plans are expected to perform as well as possible in actual operations. Therefore, when comparing the performance of different optimisation approaches, we employ a method that

incorporates actual scenarios, known as actual sample analysis. First, the optimal solution $\hat{\mathbf{y}}^*$ provided by each optimisation approach is fixed. Then, we solve the second stage problem $Q\left(\hat{\mathbf{y}}^*, \omega^{\text{actual}}\right)$, where $\omega^{\text{actual}}$ is the actual scenario. The objective value mentioned in the rest of this section refers to the $\sum_{i \in I} C_i^{\text{apron}} x_i^r + Q\left(\hat{\mathbf{y}}^*, \omega^{\text{actual}}\right)$.

Solution methods are coded in Python using the commercial MIP solver GUROBI. The KNN and RF methods applied in our study are implemented with the scikit-learn library. All experiments are conducted on a computer equipped with an INTEL CORE i7-12700K 12 Core 20 Threads CPU @ 5.00 GHz and 32 GB of memory. We set the CPU time to 3,600 seconds for each test instance.

### 3.5.2 Computational performance analysis of the BBC method

To evaluate the computational performance of the proposed BBC method, we utilise the commercial solver GUROBI to solve Model (3.8), using its results as the benchmark. We evaluate the computational performance of these solution methods on 21 test instances. In this experiment, the value of $\lambda$ is set to 100, the size of the sample set $\Omega$ is set to $|\Omega| = \{5, 10, 25, 50, 75, 100, 200, 300\}$, and the RF-CSR method is used to generate scenarios. We first compare the computational performance of the proposed BBC method and the commercial solver GUROBI. We select the optimality gap and CPU time as performance indicators, where the optimality gap indicator is computed as Equation (3.14).

$$\text{Optimality Gap } (\%) = \frac{\text{UB} - \text{LB}}{\text{UB}} * 100. \tag{3.14}$$

For brevity, we show the average computational performance of the solution method for all test instances under the same scenario number. The results are displayed in Table 3.7, where the notation "Gap" and "Time (s)" represent the mean optimality gap and the mean CPU time, respectively. As shown in Table 3.7, the overall optimality

Table 3.7: Computational performance of the BBC method for AGAP.

| Group | GUROBI | | BBC | | | |
|---|---|---|---|---|---|---|
| | Gap (%) | Time (s) | Gap (%) | Reduction (%) | Time (s) | Reduction (%) |
| 5 | 61.90 | 2,236.64 | 2.66 | 95.71 | 1,202.19 | 46.15 |
| 10 | 61.90 | 2,291.17 | 2.74 | 95.57 | 1,203.53 | 47.47 |
| 25 | 65.58 | 2,406.22 | 3.67 | 94.40 | 1,375.92 | 42.83 |
| 50 | 66.67 | 2,413.74 | 3.58 | 94.63 | 1,210.54 | 49.85 |
| 75 | 66.67 | 2,421.13 | 3.75 | 94.38 | 1,377.52 | 43.10 |
| 100 | 71.43 | 2,574.96 | 3.67 | 94.87 | 1,218.22 | 52.69 |
| 200 | 71.43 | 2,600.34 | 4.41 | 93.83 | 1,388.30 | 46.61 |
| 300 | 66.67 | 2,433.98 | 3.43 | 94.86 | 1,402.36 | 42.38 |
| Average | 66.53 | 2,422.27 | 3.50 | 94.76 | 1,282.32 | 46.44 |

gap and CPU time of the BBC method are 3.50% and 1,282.32 seconds, respectively. Compared with GUROBI, the overall optimality gap of the BBC method is sharply reduced by 94.76%, while the required CPU time is also substantially shortened by 46.44%.

Subsequently, we employ the Wilcoxon signed-rank test to examine further the differences in the optimality gap and CPU time between the BBC method and GUROBI. The results show that all P-values are less than 0.001, indicating a significant reduction in both the optimality gap and CPU time for the BBC method compared to GUROBI.

### 3.5.3 The performance of the ETO approach for AGAP

In this subsection, we first compare the performance of the ETO approach for AGAP driven by ML methods with varying prediction accuracies. This study employs KNN and RF methods to predict aircraft arrival times, noting that, as shown in Subsection 3.3.3, the prediction accuracy of RF surpasses that of KNN. In this experiment, the value of $\lambda$ is set to 100. Figure 3.6 illustrates that the ETO approach driven by the RF method outperforms the KNN approach when the scenario numbers are identical, regardless of whether the SAA or CSR method is used for scenario selection. This find-

ing suggests that the ETO approach for AGAP yields better airport gate assignment plans when the underlying ML method provides more accurate predictions.

We then investigate the influence of two scenario selection strategies mentioned in Subsection 3.4.2.2 on the ETO approach for AGAP. As shown in Figure 3.6, whether the ETO approach is based on the KNN or RF methods, we find that in most cases, using the CSR method for scenario selection is better than using the SAA method because of the smaller objective value. We also observed that when the number of cluster centres is small in the CSR method, i.e., the number of scenarios is small, better performances of the ETO approach for AGAP are provided because the obtained objective value is smaller. However, the decision performance tends to decrease as the number of cluster centres increases. This phenomenon may arise due to a decrease in cluster quality, i.e., when the number of clusters increases, some clusters may contain very few data points or even only one data point. Consequently, the quality of the cluster diminishes, rendering it ineffective in representing the underlying data structure. At the same time, it can be observed that when we use the SAA method for scenario selection, the results of the ETO approach based on the KNN or the RF methods do not show obvious characteristics. In general, the CSR method is a more appropriate scenario selection strategy because it requires fewer scenarios to achieve better performance in airport gate assignment plans, which means the CSR method helps to provide superior plans with shorter CPU times.

In previous AGAP studies, the distributions of uncertain parameters were primarily derived from historical data analysis (Şeker and Noyan, 2012; Xu et al., 2017; Kim et al., 2023a), an approach referred to as sampling from the historical data (SFHD). We further compare the performance of the ETO approach with the SFHD approach used in previous studies. The approach first analyses the aircraft arrival time distribution based on historical data, then solves the SP model of AGAP based on the sample set $\Omega$ generated by the analysed distribution. In Figure 3.5, We provide the distribution

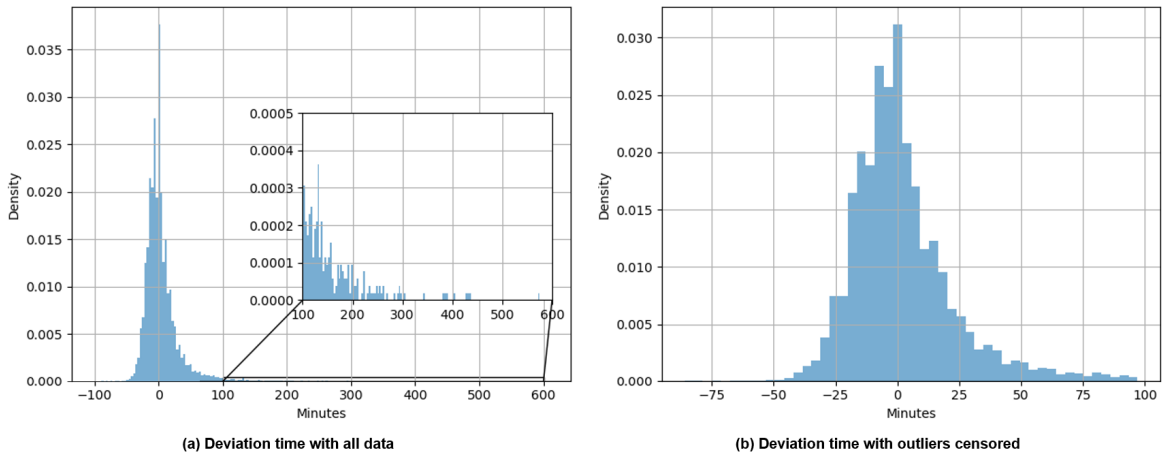


Figure 3.5: Distribution of aircraft arrival time deviations based on the historical data at XMN.

of aircraft arrival time deviations based on the historical data from September 1st to October 30th, 2023. Figure 3.5(a) presents the distribution for all deviation time data. The deviation time is concentrated around 0 minutes, but numerous extreme values result in a long-tailed distribution. Subsequently, we apply the $3\sigma$ criterion to identify values in the data that exceed three times the standard deviation, classifying them as outliers and then removing them. When considering the censored data, Figure 3.5(b) reveals a more balanced and symmetrical distribution of aircraft arrival time deviations. In the SFHD approach, the scenario $\omega \in \Omega$ can be constructed by generating aircraft arrival time deviations $\hat{a}_i$ to the estimated arrival time $A_i$, where $\hat{a}_i$ is randomly generated from the distribution of aircraft arrival time deviations provided in Figure 3.5(b).

The comparison results are provided in Figure 3.6. It can be seen that the ETO approach driven by the RF method can provide much better airport gate assignment plans than the SFHD approach when the scenario numbers are identical. Conversely, the performance of the ETO approach driven by the KNN method is similar to the SFHD approach. This suggests that when airport gate controllers have access to in-

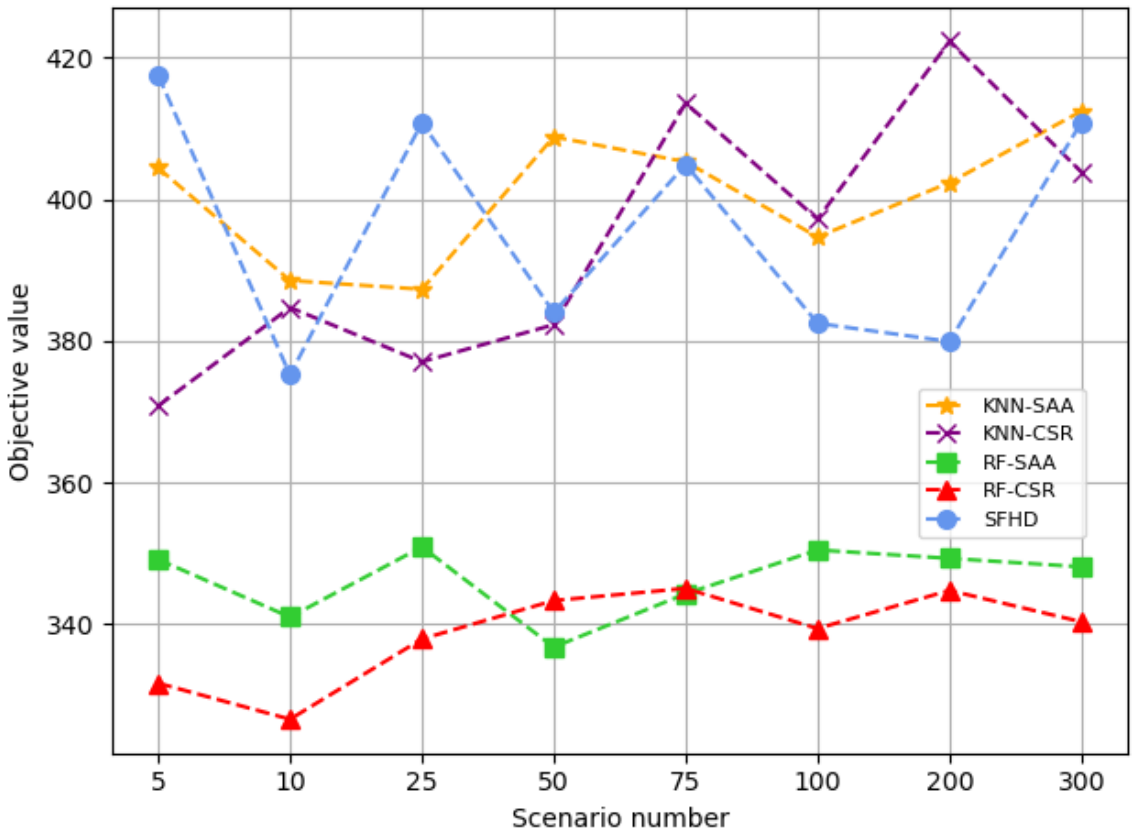


Figure 3.6: Comparison results of the ETO and the sampling from historical data approaches for AGAP.

formation other than historical arrival data, it is still necessary to properly select and use ML methods and scenario selection strategies to make the ETO approach to provide better decisions. Otherwise, the performance of the ETO approach may not be improved over the SFHD approach.

### 3.5.4 Comparison between optimisation approaches for AGAP

In this subsection, we compare the performance of various optimisation approaches for AGAP, where the value of $\lambda$ is set to 100. We use the notation DETA to represent the deterministic model with estimated aircraft arrival time for brevity. From Table 3.8, it

can be seen that the mean objective value of the ETO driven by the RF-CSR method is 326.57, which is better than other optimisation approaches. Specifically, compared to the DETA approach, the RF-CSR method achieves a 16.37% reduction in the mean objective value. Additionally, compared with PTO approaches driven by KNN and RF methods, the RF-CSR method separately reduces the mean objective value by 20.11% and 5.95%. Furthermore, compared to the SFHD approach, the RF-CSR method achieves a 12.96% reduction in the mean objective value. Lastly, the mean objective value of the RF-CSR method is reduced by 15.68%, 11.93%, and 3.03% compared to the ETO approaches based on KNN-SAA, KNN-CSR, and RF-SAA methods, respectively. We also provide the performance ranking of each optimisation approach in Table 3.8. We find that the RF-SAA method won first place the most times, 12 times. The next one is the PTO approach driven by the RF method, 9 times. The RF-CSR method, the PTO approach driven by the KNN method, and the SFHD approach tied for third place, each ranking first 8 times. Although the RF-CSR method is not the optimisation approach that gets the most first places in these test instances, it exhibits better robustness than other optimisation approaches when dealing with different test instances. This is because the rankings of the RF-CSR method fluctuate between 1 and 5, with an average ranking of 2.05. In contrast, the rankings of other optimisation approaches vary from 1 to 8, with average rankings ranging from 2.24 to 4.67. Because the RF-CSR method is more robust than other optimisation approaches in dealing with different test instances, its mean objective value is the best, and its overall rank won first place. Table 3.9 provides a detailed report on the performance of each optimisation method for every test instance.

The objective of the AGAP proposed in this study is twofold: first, to minimise the number of aircraft assigned to the apron, and second, to reduce the aircraft delay time. We employ a four-quadrant diagram to analyse the impact of different optimisation approaches on two objectives. This diagram's horizontal axis represents apron assign-

Table 3.8: A brief report on the performance of the optimisation approaches for AGAP.

| | DETA | PTO | | SFHD | ETO | | | |
|---|---|---|---|---|---|---|---|---|
| | | KNN | RF | | KNN-SAA | KNN-CSR | RF-SAA | RF-CSR |
| Mean value | 390.48 | 408.76 | 347.24 | 375.19 | 387.29 | 370.81 | 336.76 | **326.57** |
| Overall rank | 7th | 8th | 3rd | 5th | 6th | 4th | 2nd | **1st** |
| Rank 1st number | 7 | 8 | 9 | 8 | 7 | 7 | **12** | 8 |
| Best rank | **1st** | **1st** | **1st** | **1st** | **1st** | **1st** | **1st** | **1st** |
| Mean rank | 3.95 | 4.67 | 3.05 | 3.38 | 3.91 | 3.76 | 2.24 | **2.05** |
| Worst rank | 8th | 8th | 8th | 8th | 8th | 8th | 8th | **5th** |

Table 3.9: A detailed report on the performance of the optimisation approaches for AGAP.

| Instance | DETA | PTO | | SFHD | ETO | | | |
|---|---|---|---|---|---|---|---|---|
| | | KNN | RF | | KNN-SAA | KNN-CSR | RF-SAA | RF-CSR |
| 1 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| 2 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| 3 | 27 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| 4 | 364 | 387 | 373 | **299** | 339 | 347 | 351 | 350 |
| 5 | **0** | **0** | **0** | 37 | 50 | 6 | **0** | **0** |
| 6 | 378 | 404 | 317 | 341 | 423 | 414 | 308 | **284** |
| 7 | 58 | 151 | 36 | 86 | 55 | 111 | **25** | 29 |
| 8 | 325 | 395 | 396 | 324 | **280** | 318 | 389 | 292 |
| 9 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| 10 | 276 | 313 | 258 | 306 | 299 | 280 | **235** | 266 |
| 11 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| 12 | 1,227 | **923** | 960 | 1,063 | 1,071 | 1,081 | 1,040 | 1,038 |
| 13 | 165 | 193 | 146 | **79** | 142 | 141 | 99 | 88 |
| 14 | 411 | 437 | **272** | 352 | 467 | 482 | 285 | 295 |
| 15 | **483** | 666 | 686 | 583 | 662 | 527 | 783 | 556 |
| 16 | 493 | 602 | 372 | 426 | 445 | 404 | **345** | 370 |
| 17 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| 18 | 1,269 | 1,049 | 1,292 | 1,008 | 1,141 | **822** | 985 | 1,139 |
| 19 | 721 | 738 | 587 | 681 | 741 | 797 | **536** | 538 |
| 20 | 1,519 | 1,621 | **1,162** | 1,739 | 1,470 | 1,482 | 1,330 | 1,218 |
| 21 | 484 | 705 | 435 | 555 | 548 | 575 | **361** | 395 |

ment costs, while the vertical axis corresponds to aircraft delay time costs. The results are provided in Figure 3.7. The DETA and the PTO approaches driven by the KNN method fall into Quadrant I. This placement indicates that airport gate assignment

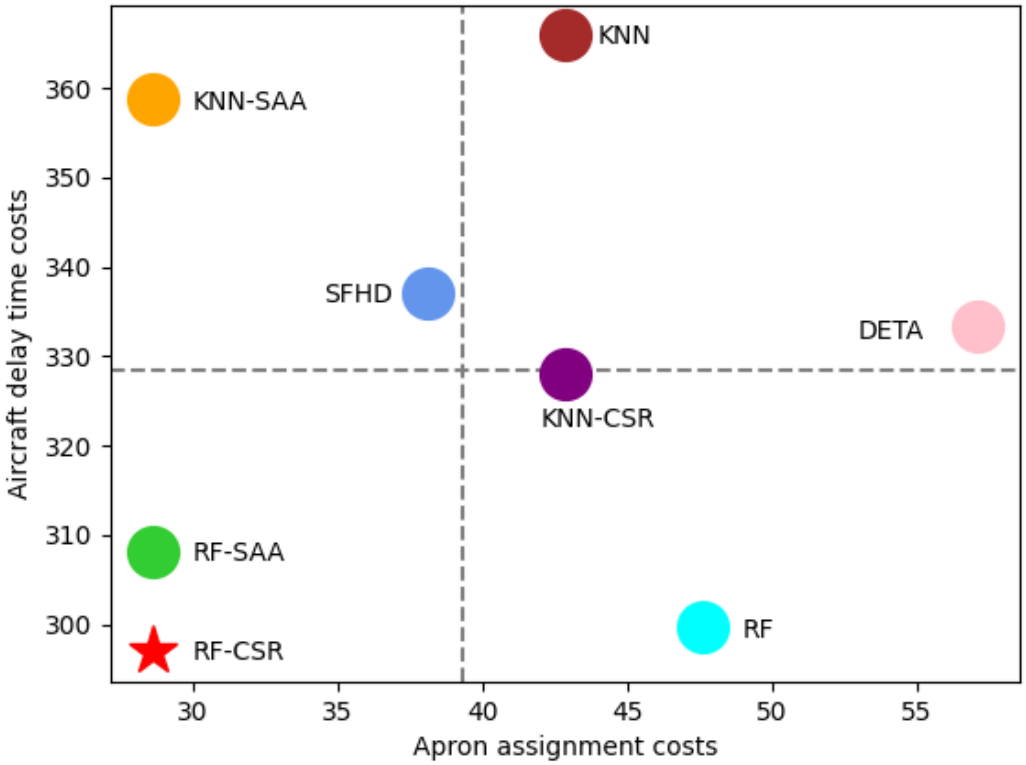


Figure 3.7: Four quadrant diagrams for the comparison of optimisation approaches.

plans generated by these approaches exhibit high apron assignment costs and high aircraft delay time costs. On the other hand, the SFHD and KNN-SAA approaches reside in Quadrant II, suggesting that they produce plans with low apron assignment costs but still suffer from high aircraft delay time costs. Conversely, the KNN-CSR method and the PTO approach driven by the RF method are located in Quadrant IV. These approaches yield decisions with high apron assignment costs but relatively low aircraft delay time costs. Finally, the RF-SAA and RF-CSR methods occupy Quadrant III, indicating that these optimisation approaches allow airport gate controllers to develop informed airport gate assignment plans with low apron assignment costs and minimal aircraft delay time costs.

Besides, this experiment reveals that the accuracy of the ML method employed in the PTO approach for AGAP significantly impacts the performance of the resulting airport gate assignment plans. Notably, when the prediction accuracy is poor, the PTO approach performs even worse than the DETA. We observed that the performance of airport gate assignment plans generated using the ETO approach has improved somewhat compared to the PTO approach. Specifically, according to the results provided in Table 3.8, KNN-SAA and KNN-CSR reduce the mean objective value of PTO based on KNN by 5.25% and 9.28%, respectively. Similarly, RF-SAA and RF-CSR achieve reductions of 3.02% and 5.95%, respectively, in the mean objective value of PTO driven by RF. Notably, regardless of the accuracy of ML methods in predicting aircraft arrival times, adopting the ETO approach leads to better-performing airport gate assignment plans. Furthermore, our research indicates that employing an effective scenario selection strategy can enhance the performance of airport gate assignment plans generated using the ETO approach. To be more specific, KNN-CSR achieves a 4.26% reduction in the mean objective value compared to KNN-SAA, while RF-CSR achieves a 3.03% reduction compared to RF-SAA.

In summary, the RF-CSR method stands out as the most effective optimisation approach. It assists airport gate controllers in formulating informed airport gate assignment plans. Specifically, this approach leads to reduced apron assignment costs and minimised aircraft delay time costs. Furthermore, the RF-CSR method exhibits better robustness when handling diverse test instances.

### 3.5.5 The impact of the subset size in the CSR method

In the above numerical experiments, we set the subset size $|\Xi^{\text{sub}}|$ in the CSR method to $10^5$ for CPU time and tractability considerations. In this subsection, we evaluate the impact of the subset size $|\Xi^{\text{sub}}|$ in the CSR method in detail. Specifically, we study the CPU time required by the CSR method when smaller or larger subset sizes $|\Xi^{\text{sub}}|$ are

used compared to $10^5$, and the performance of the gate assignment plans provided by the ETO approach based on a subset of scenarios $\Xi^{\text{sub}}$ generated by the CSR method as input. Recall that the size of the original scenario set $\Xi$ is $N^{|I|}$, which is an exponential function of $|I|$. Considering $N^{|I|}$ is an extremely large number and cannot be handled by the K-means method embedded in the CSR method within a few minutes, in this subsection, we investigate subset sizes $|\Xi^{\text{sub}}| \in \{10^3, 10^4, 10^5, 10^6, 10^7\}$.

Given that the RF-CSR method performs best in the experiments above, we conduct further experiments based on this method to evaluate the impact of subset size $|\Xi^{\text{sub}}|$. For each subset size value, we set the number of clustered scenarios to $|\Omega| \in \{5, 10, 25, 50, 75, 100, 200, 300\}$, and generate scenarios for the 21 test instances provided in the experimental design subsection 3.5.1. Therefore, for each subset size value, we generate 168 test instances. For conciseness, we present the average CPU time and objective value of all test instances for each subset size value in Figure 3.8. Our results indicate that using a larger subset size $|\Xi^{\text{sub}}|$ for the CSR method can lead to slightly better gate assignment plans. However, this improvement comes at the cost of increased CPU time required by the CSR method. Specifically, when we increase the subset size from $10^3$ to $10^7$, our average objective function indicator decreases from 342.99 to 333.07, a decrease of 2.89%. The average computing time required increases from 0.83 seconds to 1452.14 seconds, an increase of 174,947%. While larger subset sizes improve the objective function, the drastic increase in computing time suggests scalability issues that could limit the practical applicability of the CSR method for airport gate assignment planning. Determining a good subset size to balance performance improvement and acceptable computation time may be useful. Based on this consideration, choosing $10^5$ as the subset size may be a reasonable choice since the average CPU time required is 8.46 seconds, which is still reasonable, and the average objective value is 338.65, showing improvement compared to $10^3$ and $10^4$. Overall, while the increase in efficiency with larger subset sizes is promising, gate controllers

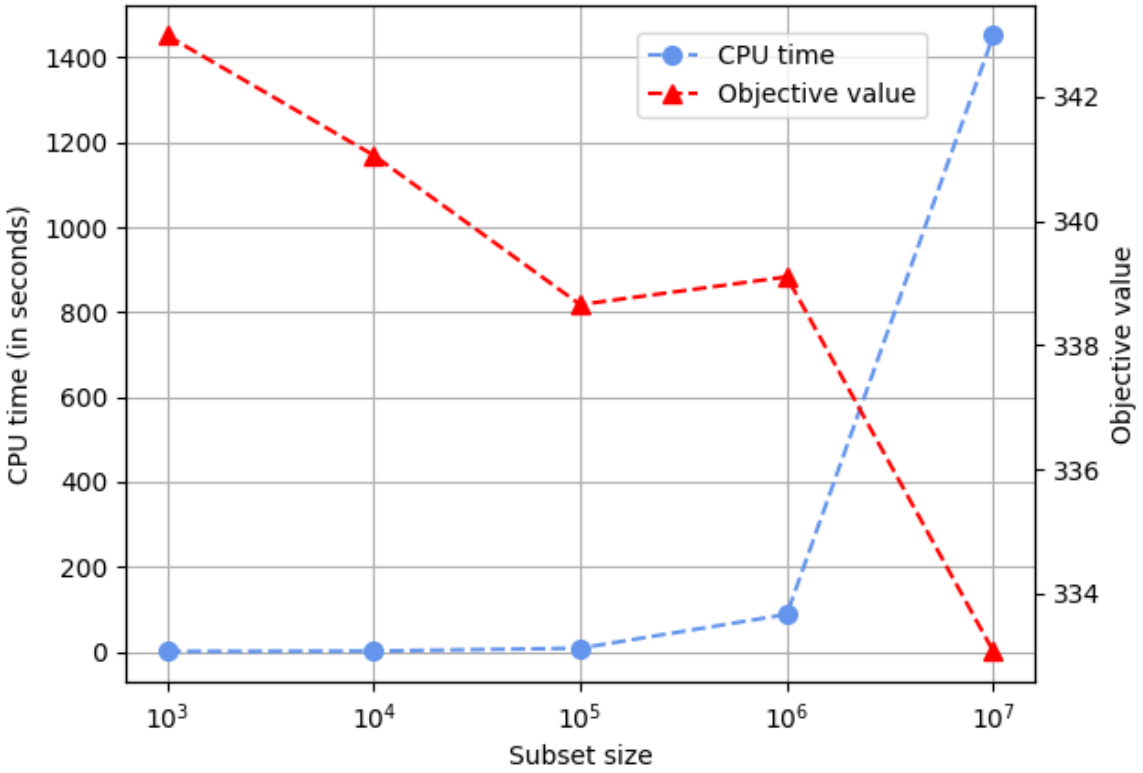


Figure 3.8: The impact of the subset size in CSR method.

should balance this against the computational cost for practical implementation.

### 3.5.6 The impact of the decision horizons for AGAP

In the experiments conducted above, the RF-CSR method emerged as the best performer among the prescriptive analytics approaches mentioned in this study. In this section, we investigate how the performance of this method varies across different decision horizons through a practical implementation experiment. The practical implementation experiment is based on a rolling horizon approach, with decision horizons set at 2-hour, 3-hour, and 4-hour, as mentioned in the experimental design subsection. Specifically, to manage the aircraft expected to arrive, the RF-CSR method for AGAP

is executed a few hours or a day in advance to obtain the gate assignment plans for each decision horizon. Let us consider the case of a 4-hour decision horizon as an example. Assuming the aircraft is scheduled to arrive at 00:00, the actual arrival time for each aircraft within this rolling horizon is determined one hour prior to 00:00. Based on this, the resource program is executed for aircraft expected to arrive between 00:00 and 04:00, determining the aircraft scheduling decisions and outputting the release time for each gate during this period. The release time $R^g$ of each gate $g \in G$ is defined as the pushback time of the last aircraft served by that gate plus the buffer time. In detail, $R^g = \min_{i \in I^g}\{\hat{t}_i + O_i + B\}$, where $I^g$ represents the set of aircraft served by the gate $g$ in the current time period, and $\hat{t}_i$ represents the parking time associated with each aircraft. The second run of the resource program occurs at 03:00. It uses the actual aircraft arrival times and gate release times $R^g$ from the previous period to make decisions for aircraft arriving between 04:00 and 08:00. The resource program then continues to operate for the remaining periods.

The recourse program mentioned above refers to making additional decisions to adapt to new information (Birge and Louveaux, 2011). In this study, we design a simple recourse program based on the second-stage problem (3.7). As illustrated in Model (3.15), we add gate release time constraints (3.15c) to ensure that each aircraft assigned to a contact gate must wait until the previous aircraft has pushed back before it can park.

$$RP\left(\mathbf{x}, \mathbf{y}, \omega^{\text{actual}}\right) = \min \quad \sum_{i \in I} C_i^{\text{delay}} d_i^{\omega^{\text{actual}}} \tag{3.15a}$$

$$\text{s.t.} \quad \text{Constraints (3.7b)-(3.7e)}, \tag{3.15b}$$

$$t_i^{\omega^{\text{actual}}} \geq R^g x_i^g, \quad \forall g \in G, \forall i \in I. \tag{3.15c}$$

Then, a practical implementation experiment is conducted using realistic data from

XMN on October 31st, 2023. We use 2-hour, 3-hour, and 4-hour decision horizons to divide the data for this day. Details of the generated test instances are provided in Table 3.6. The total costs for the 2-hour, 3-hour, and 4-hour decision horizons are 30,824, 26,486, and 21,600, respectively. This demonstrates that extending the decision horizon can better optimise resources, provide greater scheduling flexibility, and ensure a more comprehensive evaluation of gate usage and aircraft assignments, thereby providing efficient and robust gate assignment plans. However, longer decision horizons also bring computational burdens. For example, in this study, we can find the optimal solution only for the test instance with a 2-hour decision horizon. In contrast, the average optimality gaps for the test instances with 3-hour and 4-hour decision horizons are 4.57% and 4.26%, respectively. Considering that a larger decision horizon is likely to lead to a larger optimality gap, the gate controllers must balance computational efficiency and the effectiveness of the gate assignment plans when determining the decision horizon to ensure the practical applicability of the RF-CSR method in airport gate assignment planning.

### 3.5.7 The impact of the airport gate controller preference levels

This study uses the parameter $\lambda$ to represent the airport gate controllers' preference for apron assignments and aircraft delay time. In this subsection, we further evaluate the $\lambda$ within the set $\{50, 75, 100, 125, 150\}$. As the value of $\lambda$ increases, the penalty cost associated with assigning an aircraft to the apron also increases. Consequently, the airport gate controllers become more cautious about assigning aircraft to the apron and place relatively less emphasis on the aircraft delay time. Figure 3.9 shows the average number of aircraft assigned to the apron and the average aircraft delay time under different preference levels. In Figure 3.9(a), we present a briefing report. As the value of $\lambda$ increases, the average number of aircraft assigned to the apron decreases

sharply initially, followed by a slower rate of decrease until $\lambda = 150$, at which point no aircraft are assigned to the apron. Simultaneously, as the value of $\lambda$ increases, the average aircraft delay time rises rapidly in the initial stage, and subsequently, the rate of increase becomes gentle. Overall, as the value of $\lambda$ increases from 50 to 150, the average number of aircraft assigned to the apron decreases from 2.38 to 0, while the average aircraft delay time increases from 106.05 minutes to 301.19 minutes.

A detailed report involving different aircraft types is provided in Figure 3.9(b). It can be seen that the average number of E-type and D-type aircraft assigned to the apron is 0 regardless of the change of $\lambda$ value, while the average number of C-type aircraft assigned to the apron decreases from 2.38 to 0 as the value of $\lambda$ increases from 50 to 150. When it comes to the average delay time, the change in the value of $\lambda$ also has a greater impact on the C-type aircraft. As the value of $\lambda$ increases from 50 to 150, the average delay time of C-type aircraft increases monotonically from 78.14 minutes to 254.95 minutes. On the other hand, the value of $\lambda$ has a smaller impact on the average delay time of E-type and D-type aircraft. While the average delay time of E-type and D-type aircraft generally exhibits an increasing trend with the rise in $\lambda$ value, it is not strictly monotonically increasing. Specifically, when the $\lambda$ value increases from 50 to 150, the average delay time for E-type aircraft increases from 2 minutes to 3.71 minutes, with a peak value of 4.24 occurring at $\lambda = 125$. Similarly, with the $\lambda$ value increase, the average delay time for D-type aircraft increases from 25.90 minutes to 42.52 minutes, reaching a maximum of 46.38 at $\lambda = 100$. In summary, airport gate controllers face a trade-off between assigning aircraft to apron and aircraft delay time. They must navigate the delicate balance between these two objectives. Through prudent decision-making, they can assign fewer aircraft to aprons while maintaining low aircraft delay times.

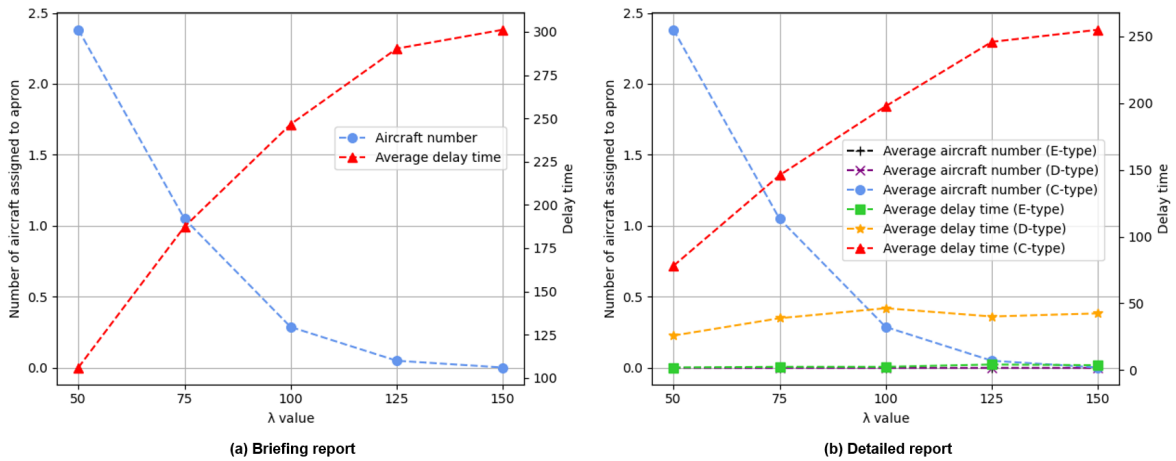


Figure 3.9: The impact of the gate controller preference levels.

### 3.5.8 Managerial implications and insights

The proposed prescriptive analytics approaches for AGAP and numerical experiments discussed above provide valuable managerial implications and insights for airport gate controllers. The following summary outlines the key findings derived from these experiments:

(i) Robust airport gate assignment plans that can alleviate congestion, absorb disruptions, and maintain high service levels are essential for maintaining a high service level at airports. Compared with other optimisation approaches, prescriptive analytics approaches, particularly the ETO approach, may offer a better solution for minimising aircraft delay time while maintaining satisfactory apron assignments under uncertain arrival times. It is recommended that airport gate controllers use historical data and prescriptive analytics approaches to more accurately predict or estimate aircraft arrival times and provide robust airport gate assignment plans.

(ii) The experimental results indicate that when information beyond historical data of uncertain parameters is available, the ETO approach driven by high-performance ML methods and appropriate scenario selection strategies can provide superior airport

gate assignment decisions compared to traditional optimisation methods. Conversely, when the prediction performance of the ML method is poor, the performance of the ETO method is similar to that of traditional optimisation approaches. This suggests that even when airport gate controllers have access to additional information, it is crucial to carefully select and develop appropriate ML methods and scenario selection strategies to enable the ETO approach to deliver better airport gate assignment plans. Otherwise, the ETO approach may not yield significant performance improvements. Furthermore, we compare the impact of decision horizons. While extending the decision horizon of the ETO approach can provide better gate assignment plans, it will likely lead to a larger optimality gap. Therefore, gate controllers must balance computational efficiency and the effectiveness of gate assignment plans when determining the decision horizon. This balance is essential to ensure the practical applicability of the ETO approach in airport gate assignment planning.

(iii) We developed a CSR method for scenario selection. Compared to the widely used SAA method, the CSR method requires fewer scenarios to perform better in airport gate assignment plans, resulting in superior plans with shorter CPU times. Additionally, the CSR method demonstrates better robustness when handling different test instances. Consequently, the CSR method generally provides satisfactory airport gate assignment plans. Our results indicate that using a larger subset size $|\Xi^{\text{sub}}|$ can lead to slightly better gate assignment plans; however, this improvement comes at the cost of significantly increased CPU time. Therefore, determining a good subset size in the CSR method to balance performance improvement and acceptable computation time may be beneficial.

(iv) Experiments also demonstrate that the proposed exact BBC method significantly outperforms the commercial solver in terms of computational performance. This enables airport gate controllers to utilise the ETO approaches to effectively capture the dynamic fluctuations of aircraft arrival times over the entire planning horizon. This

allows for the rapid development of effective and robust airport gate assignment plans.

## 3.6 Conclusions

In this study, we study the AGAP, which aims to optimise the aircraft-to-gate assignments, aircraft sequence for each contact gate, and aircraft scheduling plans to minimise the number of aircraft assigned to the apron and aircraft delay times. The practical aircraft assignment requirements are taken into account. Two prescriptive analytics approaches are developed for AGAP to address the uncertainty of aircraft arrival times, where the KNN and RF methods are used. The PTO approach assumes uncertain aircraft arrival times are identical to the values predicted by ML methods. However, the accuracy of predictions impacts the assignment plans provided by this approach. The ETO approach is further employed to address this issue, which estimates the distribution of uncertain arrival times and solves the SP model for AGAP based on this distribution. The ETO approach generally yields superior airport gate assignment decisions but requires generating numerous scenarios based on distributions, leading to scalability issues. Therefore, we developed a CSR method as a scenario selection strategy and proposed an exact BBC method to mitigate this challenge. Furthermore, we conduct a series of numerical experiments based on real-world data from XMN to demonstrate the effectiveness of the prescriptive analytics approaches and BBC algorithm. We compare the computational performance of the BBC method against a commercial solver, with results showing highly significant statistical improvements in the proposed BBC method over the benchmark method. After analysing the impact of ML methods and scenario selection strategies on the ETO approach, we compare the performance of airport gate assignment plans generated by prescriptive analytics approaches with those produced by other optimisation approaches. Notably, the ETO approach outperforms the PTO approach driven by the same ML method. Further-

more, we find that the ETO approach, supported by high-performance ML methods and effective scenario selection strategies, provides better performance airport gate assignment plans than current optimisation approaches, making it more effective in real-world applications of airport gate assignment. Finally, we investigate the impact of airport gate controller preference levels and propose some valuable managerial implications and insights that are practically useful for the decision-making of airport gate controllers.

One limitation of this study is the small size of the current dataset. Consequently, we chose KNN and RF methods, which perform better on small datasets, to drive the prescriptive analytics approaches. In future research, we will develop ML methods that implement complex structures with the support of larger datasets, enabling us to provide more accurate predicted values or estimated distributions as input for subsequent optimisation methods. Another limitation is that in the ETO approach for AGAP, the distribution of uncertain parameters provided by the ML methods is assumed to be the true distributional information and is directly used in the SP model. However, ML methods inherently produce prediction errors that cannot be entirely eliminated, meaning the distributions of uncertain parameters provided by these methods may be inaccurate. Combining the ETO approach with distributionally robust optimisation may lead to better results, particularly when the distributional information is misspecified. Furthermore, this study utilises prescriptive analytics approaches for AGAP, and extensive numerical experiments demonstrate the effectiveness of these approaches in generating high-performance airport gate assignment plans. Therefore, we recommend extending the implementation of prescriptive analytics approaches to other airside operations or airport decision-making problems. By integrating operations research techniques with ML methods, airside controllers can enhance operational efficiency, increase capacity, alleviate congestion, manage disruptions, and maintain a high service level at the airside.

# Chapter 4

# Prescriptive analytics for the aircraft sequencing and scheduling problem

## 4.1 Introduction

With the continuous increase in air traffic demand in recent years, the airspace systems of some countries or regions are operating near their capacity (Ng et al., 2017; Solak et al., 2018; Ribeiro et al., 2019). In an environment with limited capacity, any reduction in system capacity can lead to significant delays and substantial losses for airlines and passengers, posing severe challenges to the air traffic management (ATM) system (Solak et al., 2018; Khassiba et al., 2020). Runways are considered one of the primary bottlenecks of the ATM system and play a crucial role in determining the capacity of both airports and airspace systems (Balakrishnan and Chandran, 2010; Ghoniem et al., 2014; Ikli et al., 2021). Due to the high investment costs and prolonged construction periods, building new infrastructure, such as runways and airports, to increase the capacity of airspace systems is often not an immediate solution (Ikli et al., 2021). An

alternative promising approach to enhancing airspace system capacity is to improve the utilisation of existing runways (Ng et al., 2017; Solak et al., 2018; Ikli et al., 2021). The optimisation problem studied to optimise runway utilisation is known as the RSP (Bennell et al., 2011; Ikli et al., 2021). It can be further divided into three categories: the ALP for arriving aircraft, the ATP for departing aircraft, and the ASSP for both arrival and departure aircraft (Bennell et al., 2011; Ng et al., 2017; Ikli et al., 2021; Messaoud, 2021). The RSP determines the sequence and schedule of landing and taking off aircraft to optimise predefined objectives while adhering to various operational constraints. These constraints generally include spacing, wake vortex separation requirements, and operational time windows (Bennell et al., 2011; Solak et al., 2018; Ikli et al., 2021). Two main objective functions for the RSP are discussed in the literature, including minimising the makespan of the runway system (Balakrishnan and Chandran, 2010; Harikiopoulo and Neogi, 2010; Prakash et al., 2021) and minimising the total, average, or weighted delay of all aircraft (Sama et al., 2017; Pohl et al., 2021; Prakash et al., 2022; Pohl et al., 2022).

The classical RSP, in which all input information is assumed to be deterministic, has been well studied (Bennell et al., 2011; Solak et al., 2018; Ikli et al., 2021; Messaoud, 2021). However, with increasing air traffic and factors such as severe weather, delay propagation, probabilistic nature of trajectories, technical difficulties, and security concerns, aircraft arrival and departure times have become increasingly uncertain (Ng et al., 2017; Solak et al., 2018; Khassiba et al., 2020, 2022; Kim et al., 2023a). Uncertain parameters can render predetermined aircraft sequencing and scheduling plans ineffective. Consequently, developing robust plans to address these uncertainties has become a key focus in recent runway operations research. Solveling et al. (2011) developed an SP approach for ASSP under the uncertainty of arrival and departure times in a parallel runway system, where one runway is designated for arrivals and the other for departures. Their results demonstrate the potential benefits of the SP model

over a deterministic model during peak hours. Ng et al. (2017) investigated the ASSP under uncertainty in a mixed-mode parallel runway system. The study employed an RO approach to develop aircraft sequencing and scheduling plans for worst-case scenarios. Solak et al. (2018) introduced SP models based on network and slot formulations for ASSP under uncertainty. Experiments were conducted in a parallel runway system with two independently operated runways, one for arrivals and one for departures. The results demonstrate that the proposed SP models are practically implementable and offer potential advantages over deterministic models. Khassiba et al. (2020) proposed an SP model with chance constraints for the extended ALP (EALP) on a single landing runway. This model aims to pre-schedule aircraft at a destination airport to minimise landing sequence length and time-deviation costs. The validation with realistic data from Charles-de-Gaulle (CDG) airport demonstrates its advantages over deterministic policies. Khassiba et al. (2022) extends previous research (Khassiba et al., 2020) on the EALP under uncertainty by incorporating multiple initial approach fixes (IAFs) and different initial aircraft statuses. The study introduces two problem variants based on IAF assignment flexibility and uses realistic data from CDG airport to demonstrate the benefits of the SP models and IAF re-assignment.

The aforementioned studies on stochastic optimisation for runway operations primarily focus on international airports with multiple runways. At these airports, each runway is often dedicated exclusively to either arrivals or departures to reduce conflicts and improve operational efficiency. Consequently, these studies mainly examine the operational efficiency of single or dual runways dedicated to independently managing arrivals and departures. While many international airports are equipped with multiple runways to meet high air traffic demand, there are also numerous international airports with only one runway that must handle a large number of aircraft. Examples include San Diego International Airport (SAN), London Stansted Airport (STN), London Gatwick Airport (LGW), Xiamen Gaoqi International Airport (XMN), Urumqi

Diwopu International Airport (URC), Fukuoka Airport (FUK), and Chhatrapati Shivaji Maharaj International Airport (BOM), etc. Given that single-runway airports have limited resources and must handle both arriving and departing aircraft, the uncertain arrival and departure times can significantly impact the scheduled aircraft sequencing and scheduling plans. However, relevant research on mixed-operation single-runway operations under uncertainty is quite scarce. Therefore, this study addresses this research gap by exploring efficient and robust aircraft sequencing and scheduling plans for single-runway airports under uncertainty.

Additionally, in previous studies, stochastic optimisation approaches for runway operations primarily derive the probability distributions of uncertain arrival and departure times from the analysis of historical data (Solveling et al., 2011; Solak et al., 2018) or empirical knowledge (Ng et al., 2017; Khassiba et al., 2020, 2022). With the development of big data technology, massive amounts of data provide new opportunities for addressing uncertainty in runway operations. ML methods can be utilised to more accurately estimate the probability distributions of arrival and departure times when airport runway controllers have access to historical and auxiliary data (Tian et al., 2023c; Wang and Yan, 2023). Driven by these estimations, subsequent optimisation methods have the potential to provide runway scheduling plans that are more closely aligned with the actual conditions on the day of operation. The integration of predictive and optimisation methods to make informed decisions based on available data is known as prescriptive analytics (Bertsimas and Kallus, 2020; Qi and Shen, 2022; Wang and Yan, 2023; Tian et al., 2023a,b,c).

In this study, we employ the ETO approach in prescriptive analytics for ASSP. This approach involves initially estimating the probability distributions of uncertain parameters using ML methods, followed by solving an SP model based on these estimated probability distributions (Bertsimas and Kallus, 2020; Qi and Shen, 2022; Yan et al., 2022; Yang et al., 2024; Wang et al., 2024). In the ETO approach for ASSP,

we first use the RF method to estimate the probability distributions of aircraft arrival and departure times. The RF method is a supervised learning technique that uses multiple decision trees for prediction. Due to its high prediction accuracy, robustness to outliers, and relatively good interpretability, the RF method has also been widely used in other prescriptive analytics studies (Bertsimas and Kallus, 2020; Yan et al., 2020; Galli et al., 2021; Yan et al., 2024a). Appropriate scenario selection strategies are then employed to sample a suitable number of scenarios from the estimated aircraft arrival and departure time probability distributions, serving as input to solve the SP model for ASSP. In practical runway operations, runway controllers can establish aircraft sequencing decisions tens of minutes in advance (Solak et al., 2018; Pohl et al., 2021). Subsequently, as more specific information on uncertain parameters becomes available, they make scheduling arrangements (Solak et al., 2018; Khassiba et al., 2020, 2022). Therefore, we propose an SP model featuring a two-stage decision-making process for ASSP. In the first stage, aircraft sequencing decisions are made prior to the realisation of uncertainties. In the second stage, aircraft scheduling decisions are made after observing the arrival and departure times.

In the ETO approach, the probability distributions of uncertain parameters provided by the ML method are assumed to represent the true probability distributions. However, ML methods inherently produce prediction errors that cannot be completely eliminated. Additionally, when using scenario selection strategies to sample probability distributions, the results may deviate from the true probability distributions due to outliers, inadequate scenario coverage, and other factors. Overlooking these potential prediction and sampling errors may result in decisions that perform unsatisfactorily in real-world scenarios. We propose a novel prescriptive analytics framework called the ETDRO approach to address these issues. Like the ETO approach, an ML method provides the probability distributions of uncertain parameters. However, in the subsequent optimisation, the ETDRO approach addresses prediction and sampling

errors by developing a DRO model, assuming that the true probability distributions are completely unknown or only partially known. The estimated probability distributions of the uncertain parameters, provided by the ML method, are used as the reference probability distributions in the subsequent DRO model. The DRO models aim to optimise the expected value of a given function for the worst-case probability distributions within an ambiguity set. The effectiveness of these models largely depends on the selection of ambiguity sets. In this research, we utilise the type-1 Wasserstein ambiguity set, which offers several advantages: it requires only minimal sampling data for construction, allows decision-makers to adjust the level of aversion to ambiguity by modifying the radius of the Wasserstein ball, and demonstrates strong convergence characteristics (Mohajerin Esfahani and Kuhn, 2018; Bansal et al., 2018; Zhang et al., 2021; Agra and Rodrigues, 2022; Shehadeh, 2023). Calculating the type-1 Wasserstein distance between two distributions can be framed as a transportation problem, where the objective is to transfer the probability mass from one distribution to the other (Mohajerin Esfahani and Kuhn, 2018; Bansal et al., 2018; Agra and Rodrigues, 2022).

While DRO approaches are more flexible and robust than traditional SP approaches, especially when data are incomplete or high uncertainty, they are often challenging to solve. Consequently, the DRO model typically requires additional reformulation or decomposition methods to ensure tractability (Mohajerin Esfahani and Kuhn, 2018; Bansal et al., 2018; Zhang et al., 2021; Shehadeh, 2023). In this study, we ensure the computational tractability of the DRO model for ASSP embedded in the ETDRO approach by utilising the distributionally robust (DR) L-shaped method proposed by Bansal et al. (2018). The DR L-shaped method, an exact solution method, is widely applied in research on DRO approaches across transportation, logistics, supply chain, the service industry, and power systems (Guevara et al., 2020; Liu et al., 2021; Agra and Rodrigues, 2022; Gangammanavar and Bansal, 2022; Wang et al., 2022b; Black et al., 2023; Xu et al., 2024). Given the challenges in solving DRO models for optimisation

problems related to scheduling decisions, we propose an enhanced DR (E-DR) L-shaped method incorporating several algorithmic enhancements. These enhancements involve the inclusion of time constraints (Agra and Rodrigues, 2022; Wang et al., 2022b), LBL cuts (Adulyasak et al., 2015; Wu et al., 2022; Shehadeh, 2023; Tsang et al., 2024), and initial optimality cuts (Zhang et al., 2021; Gong and Zhang, 2022; Yin et al., 2024). Additionally, solving the master problems of the DR L-shaped and E-DR L-shaped methods for the DRO model of the ASSP is highly time-consuming due to their mixed-integer programming nature (Rei et al., 2009; Rahmaniani, Ragheb and Crainic, Teodor Gabriel and Gendreau, Michel and Rei, Walter, 2017; Rahmaniani et al., 2018). Including cuts in each iteration further increases the complexity of the master problems (Rei et al., 2009). To mitigate the time-consuming issue of solving the MILP master problems in each iteration, we introduce the BBC method within the framework of the E-DR L-shaped method. Rather than solving the complex master problems in each iteration, the BBC method constructs a single branch-and-bound tree, adding Benders cuts to unfathomed nodes after identifying integer first-stage solutions during the branch-and-bound search (Gendron et al., 2016). Based on the BBC method, we further propose the E-DR-BBC method for the DRO model of the ASSP.

The exact E-DR L-shaped and E-DR-BBC methods proposed in this paper can enhance computational performance compared to the previous DR L-shaped method. Nevertheless, the ASSP is an NP-hard problem with near real-time requirements (Bennell et al., 2011; Ikli et al., 2021; Chen et al., 2024). Consequently, these exact decomposition methods may not deliver the optimal solution for the DRO model of the ASSP within a reasonable timeframe when addressing large-scale test instances. As the DRO approach is increasingly applied to more complex optimisation problems, inexact solution methods are also increasingly utilised to find acceptable, high-quality solutions within a limited timeframe (Zhang et al., 2021; Gangammanavar and Bansal, 2022; Zhang et al., 2023; Tsang et al., 2023, 2024). Therefore, we further develop an

inexact decomposition approach for the DRO model of the ASSP, based on the constrained position shifting (CPS) method proposed by (Balakrishnan and Chandran, 2010). Specifically, we stabilise the master problem of the E-DR-BBC method by utilising CPS constraints to search for new solutions with high quality around a stability centre point (i.e., a good feasible first-stage solution) (Baena et al., 2020; Prakash et al., 2021, 2022; Gong and Zhang, 2022). Consequently, this proposed inexact solution method is termed the stabilised DR-BBC (S-DR-BBC) method. It is worth noting that while adding CPS constraints shortens CPU time by reducing the search space, it also eliminates part of the feasible domain of the original problem, thereby losing the optimality guarantee for the S-DR-BBC method.

The main contributions of this study are summarised as follows:

(i) We develop the ETO approach for ASSP under the uncertainty of aircraft arrival and departure times at a single-runway airport. The initial step involves estimating the probability distributions of these uncertain times, followed by solving the SP model for ASSP based on these estimations. The well-constructed and validated RF method is utilised to estimate these probability distributions. To the best of our knowledge, this is the first implementation of the ETO approach in the research of runway operations.

(ii) We propose a novel ETDRO approach that incorporates prediction and sampling errors into the decision-making process. In this approach, the true probability distributions are assumed to be completely unknown or only partially known. The estimated probability distributions provided by the RF method serve as reference distributions in the subsequent DRO model. This aims to mitigate the potential prediction and sampling errors associated with ML methods, thereby enhancing the efficiency and robustness of the decisions.

(iii) Although the ETDRO approach generally leads to better decisions, solving the DRO model poses a significant scalability challenge. To address this issue, we propose an E-DR L-shaped method for the DRO model of the ASSP, incorporating

algorithmic enhancements such as time constraint inclusion, LBL cuts, and initial optimality cuts. Additionally, we introduce the E-DR-BBC method, based on the BBC method, to solve the MILP master problem only once, thereby saving substantial CPU time. Furthermore, we develop the inexact S-DR-BBC method, which leverages CPS constraints to enhance computational efficiency while ensuring high-quality solutions.

(iv) We evaluate the performance of the proposed ETO and ETDRO approaches for ASSP, as well as the exact and inexact decomposition methods, using real data from XMN, a major airport on the southeastern coast of China and the busiest single-runway airport in that country. Specifically, after analysing the performance of the ETDRO approach under different Wasserstein radii, we compare the aircraft sequencing and scheduling plans generated by the ETDRO approach with those generated by other optimisation approaches, demonstrating the efficiency and robustness of the ETDRO approach. Furthermore, comparing the proposed exact and inexact decomposition methods with the DR L-shaped method, we find that the computational performance is significantly improved while maintaining high-quality solutions. This improvement is beneficial for adopting the ETDRO approach in real-world runway operations.

The remainder of this chapter is organised as follows. Section 4.2 provides the ETO approach for ASSP. Section 4.3 further illustrates the ETDRO approach for ASSP. The exact and inexact decomposition methods are proposed to solve the DRO model for ASSP efficiently. In Section 4.4, we perform numerical experiments using real-world data from XMN. The results of the scalability analyses are reported in Section 4.5. Finally, the conclusions are presented in Section 4.6.

## 4.2 ETO approach for ASSP

This section proposes the ETO approach for ASSP at single-runway airports. In Subsection 4.2.1, we present the experimental data and several selected features, fine-tune

the RF method's hyperparameters and evaluate its performance in predicting aircraft arrival and departure times. Based on these predictions, we estimate the distributions of uncertain parameters and generate the scenario set in Subsection 4.2.2. Finally, in Subsection 4.2.3, we introduce the SP model with a two-stage decision-making process for ASSP.

### 4.2.1 RF method for predicting aircraft arrival and departure times

Directly predicting aircraft arrival and departure times can be influenced by various uncertain factors. By instead predicting the deviation between estimated and actual arrival and departure times, we can more effectively account for these uncertainties and improve the accuracy of our predictions. Additionally, estimated arrival and departure times are usually readily available in historical data, allowing us to leverage this existing information to enhance our predictive capabilities. Specifically, we predict the deviation value and then add this predicted deviation value, $\hat{a}_i$, to the estimated arrival and departure time, $T_i$, of aircraft $i$ to obtain the predicted arrival and departure time $\hat{T}_i$, i.e., $\hat{T}_i = T_i + \hat{a}_i$.

The RF method applied in our study is implemented with the scikit-learn library. The dataset used in this study was collected from XMN between 1st September and 31st October 2023. It contains 30,104 records of arriving and departing aircraft, each with relevant aircraft information. The TAF data include details such as maximum and minimum temperatures, humidity, air pressure, wind direction, wind speed, and other relevant parameters during the forecast period. We selected 15 features from the original dataset, which are presented in Table 4.1 along with their data type, encoding method, and statistical information. We identified missing values in features such as "Aircraft type", "Route distance", and "Fuel load". To address these, we filled in the missing values using the median for "Aircraft type" and the mean for

Table 4.1: Features of RF for ASSP.

| Feature name | Data type | Encoding | Null count |
|---|---|---|---|
| Aircraft type | Object | Label encoding | 123 |
| Domestic/International | Object | One-hot encoding | 0 |
| Estimated departure time | Numerical | | 0 |
| Estimated arrival time | Numerical | | 0 |
| Cruise speed | Numerical | | 163 |
| Straight line distance | Numerical | | 0 |
| Route distance | Numerical | | 1752 |
| Estimated flight time | Numerical | | 0 |
| Fuel load | Numerical | | 1814 |
| High temperature | Numerical | | 0 |
| Low temperature | Numerical | | 0 |
| Humidity | Numerical | | 0 |
| Barometer | Numerical | | 0 |
| Wind direction | Object | One-hot encoding | 0 |
| Wind speed | Numerical | | 0 |

both "Route distance" and "Fuel load". Notably, the features "Aircraft type", "Domestic/International", and "Wind direction" are expressed in a literal format, necessitating their conversion to numerical data. Regarding aircraft type, the arriving aircraft at XMN primarily fall into two categories: heavy and large, which we encoded as 2 and 1, respectively. We considered 16 wind directions and adopted one-hot encoding for wind directions. We designated the data collected between 1st September and 30th October 2023 as dataset $D$ with 29,604 records of arriving and departing aircraft. We randomly split 80% of the data in $D$ as a training set $D^{\text{Training}}$ for training the RF model, while the remaining 20% as $D^{\text{Testing}}$ to evaluate its performance. The data from 31st October 2023 is used to generate test instances for the numerical experiments and scalability analysis.

In the RF method, several key hyperparameters must be considered. The hyperparameters tuned for the RF method are detailed in Table 4.2. These optimal hyperparameter values are determined using a grid search method with 5-fold cross-validation

Table 4.2: Best hyperparameter values of RF for ASSP.

| Hyperparameters | Search space | Best value |
|---|---|---|
| *n_estimators* | [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000] | 700 |
| *min_samples_split* | [2, 3, 4, 5, 6, 7, 8, 9, 10] | 10 |
| *min_samples_leaf* | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] | 2 |
| *max_depth* | [10, 20, 30, 40, 50, 60, 70, 80, 90, 100] | 80 |

Table 4.3: Prediction results on the testing dataset for ASSP.

| Metrics | Benchmark | RF |
|---|---|---|
| MAE | 35.62 | 14.38 |
| MSE | 28310.35 | 741.05 |
| RMSE | 168.26 | 27.22 |

on the training dataset $D^{\text{Training}}$.

After predicting the deviation value $\hat{a}_i$ using the ML methods, we calculate the predicted arrival and departure time $\hat{T}_i$ using the formula $\hat{T}_i = T_i + \hat{a}_i$, where the estimated arrival and departure time $T_i$ is assumed to be known in advance. We further use the estimated arrival and departure time $T_i$ to replace the predicted arrival and departure time $\hat{T}_i$ in Equation (3.2) to (3.4) for calculating the MAE, MSE, and RMSE values as benchmarks. Based on the prediction results presented in Table 4.3, the RF method is better than the benchmark, as it yields smaller MAE, MSE, and RMSE values.

### 4.2.2 Distribution estimation and scenario generation

After training, testing and validating the RF method, we utilise the prediction results to more accurately estimate distributions for aircraft arrival and departure times. Set $N$ as the predetermined number of decision trees. That is, there exist $N$ possible values of the arrival and departure time of aircraft $i$ provided by the RF method. We then denote $\mathcal{P}(f_i)$ as the set containing $N$ possible values when using input feature vector $f_i$. The vector of the arrival and departure time for all aircraft in

$I$ can be written as $\widetilde{\boldsymbol{T}} = \left(\widetilde{T}_1, \widetilde{T}_2, \ldots, \widetilde{T}_{|I|}\right)$. And the Cartesian product $\Phi\left(\widetilde{\boldsymbol{T}}\right) = \left\{\mathcal{P}\left(f_1\right) \times \mathcal{P}\left(f_2\right) \times \cdots \times \mathcal{P}\left(f_{|I|}\right)\right\}$ can be used to approximate the distribution of $\widetilde{\boldsymbol{T}}$ (Yan et al., 2022; Yang et al., 2024). The Cartesian product contains $N^{|I|}$ elements, which is an exponential function of $|I|$ (Yan et al., 2022). Based on this, the uncertainty of aircraft arrival and departure times is represented by a finite set $\Xi = \left\{1,\ 2,\ ...,\ \left|\Phi\left(\widetilde{T}\right)\right|\right\}$ of scenarios, where the size of this set is also $N^{|I|}$. Considering all scenarios in $\Xi$ would render the subsequent SP model too complex to solve. Therefore, we adopt the SAA method to approximate the true distribution of aircraft arrival and departure times by randomly selecting a subset of scenarios, $\Omega$, from $\Xi$.

### 4.2.3 SP model for ASSP

In this subsection, we formulate the ASSP under arrival and departure time uncertainty as an SP model. The SP model involves a two-stage decision-making process. In the first stage, aircraft sequencing decisions are made before the uncertainties are realised. In the second stage, aircraft scheduling decisions are made after observing the arrival and departure times. In this study, we assume that the uncertainties in aircraft arrival and departure times can be approximated by a potentially small and finite scenario set $\Omega = \{1, 2, \ldots, |\Omega|\}$, comprising $|\Omega|$ independent scenarios generated by the RF method.

The SP approach assumes that the probability of each scenario is known, and the objective function is optimised based on the expected value (Birge and Louveaux, 2011). In the SP model for ASSP, we assume that each scenario $\omega \in \Omega$ is associated with a reference scenario probability $\rho^{\omega}$. Consider a set $I$ that contains arriving and departing aircraft. The variable $y_{ij}$ equals 1 if aircraft $i \in I$ precedes aircraft $j \in I \setminus \{i\}$ in the runway sequence, though not necessarily immediately. For each aircraft $i$, we define an estimated arrival or departure time $T_i^{\omega}$ under scenario $\omega \in \Omega$. The landing or take-off time $t_i^{\omega}$ of aircraft $i$ under scenario $\omega$ should be no earlier than its estimated time $T_i^{\omega}$. The delay time $d_i^{\omega}$ of aircraft $i$ under scenario $\omega$ is defined as the difference between $t_i^{\omega}$

Table 4.4: Notations and definitions for ASSP.

| Notation | Definition |
|---|---|
| **Sets** | |
| $I$ | The set of arriving and departing aircraft. |
| **Parameters** | |
| $T_i^\omega$ | Estimated arrival or departure time of aircraft $i$ under scenario $\omega$. |
| $S_{ij}$ | The separation time between aircraft $i$ and $j$. |
| $A^\omega$ | The target time of the earliest arriving and departing aircraft under scenario $\omega$. |
| $C_i^{\text{delay}}$ | Unit time delay cost for aircraft $i$. |
| $W_1$ | The weight of the makespan. |
| $W_2$ | The weight of the aircraft delay costs. |
| $M$ | A sufficiently large number. |
| **Variables** | |
| $y_{ij}$ | 1, if aircraft $i$ precedes aircraft $j$ in the runway sequence, though not necessarily immediately; 0, otherwise. |
| $t_i^\omega$ | Landing or take-off time of aircraft $i$ under scenario $\omega$. |
| $z^\omega$ | Makespan under scenario $\omega$. |
| $d_i^\omega$ | Delay time of aircraft $i$ under scenario $\omega$. |

and $T_i^\omega$. The runway can accommodate only one aircraft at a time, separation time $S_{ij}$ is required to ensure safety when aircraft $i$ and $j$ use the runway consecutively. The target time of the earliest arriving and departing aircraft under scenario $\omega$ is $A^\omega$, where $A^\omega = \min_{i \in I} \{T_i^\omega\}$. The variable $z^\omega$ represents the makespan under scenario $\omega$, which is defined as the difference between the scheduled time of the last aircraft to use the runway and $A^\omega$.

The objective function of the SP model for ASSP is to minimise the expected makespan and the expected costs caused by aircraft delays. The notation $C_i^{\text{delay}}$ represents the unit time delay cost for aircraft $i$. In this study, weights $W_1$ and $W_2$ are used to represent the runway controllers' preference levels for makespan and aircraft delay costs, respectively, where $W_1 + W_2 = 1$. For brevity, we provide notations of sets, parameters and decision variables used in Table 4.4.

Based on the problem description and notations provided above, the SP model for ASSP is provided as follows:

$$\min \ \ \mathbb{E}_{\omega}\left[Q\left(\boldsymbol{y}, \omega\right)\right] \tag{4.1a}$$

$$\text{s.t.} \quad y_{ij} + y_{ji} = 1, \quad \forall i \in I, \ \forall j \in I, \ i \neq j, \tag{4.1b}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i \in I, \ \forall j \in I, \ i \neq j, \tag{4.1c}$$

where the term $\mathbb{E}_{\omega}\left[Q\left(\boldsymbol{y}, \omega\right)\right]$ represents the expected value of the second-stage objective function, which can also be expressed as $\sum_{\omega \in \Omega} \rho^{\omega} Q\left(\boldsymbol{y}, \omega\right)$. Here, $\rho^{\omega}$ denotes the probability of scenario $\omega$, and $Q\left(\boldsymbol{y}, \omega\right)$ represents the optimal value of the second-stage problem under scenario $\omega$. For a feasible first-stage solution $\boldsymbol{y}$ and realisation of scenario $\omega$, the term $Q\left(\boldsymbol{y}, \omega\right)$ is formulated as follows:

$$Q\left(\boldsymbol{y}, \omega\right) = \min \ \ W_1 z^{\omega} + W_2 \sum_{i \in I} \left(C_i^{\text{delay}} d_i^{\omega}\right) \tag{4.2a}$$

$$\text{s.t.} \quad t_i^{\omega} \geq T_i^{\omega}, \quad \forall i \in I, \tag{4.2b}$$

$$t_i^{\omega} + S_{ij} - t_j^{\omega} \leq M(1 - y_{ij}), \quad \forall i \in I, \ \forall j \in I, \ i \neq j, \tag{4.2c}$$

$$z^{\omega} \geq t_i^{\omega} - A^{\omega}, \quad \forall i \in I, \tag{4.2d}$$

$$d_i^{\omega} \geq t_i^{\omega} - T_i^{\omega}, \quad \forall i \in I, \tag{4.2e}$$

$$t_i^{\omega} \in \mathbb{R}^{+}, \quad \forall i \in I, \tag{4.2f}$$

$$z^{\omega} \in \mathbb{R}^{+}, \tag{4.2g}$$

$$d_i^{\omega} \in \mathbb{R}^{+}, \quad \forall i \in I. \tag{4.2h}$$

The Objective function (4.1a) minimises the expected second-stage recourse costs. Constraints (4.1b) establish the landing and take-off sequences for the aircraft. The Objective function (4.2a) minimises the weighted sum of the makespan and the total aircraft delay costs under scenario $\omega$. Constraints (4.2b) require that the landing or take-off time $t_i^\omega$ of aircraft $i$ should be greater than or equal to its estimated time $T_i^\omega$. It should be noted that we do not impose a UB on the scheduled time for each aircraft. This is because arriving or departing aircraft may occasionally experience a prolonged hold before landing or take-off, although this is rare. Constraints (4.2c) ensure the separation time requirements between two aircraft. Constraints (4.2d) compute the makespan. Constraints (4.2e) determine the delay time of each aircraft. Constraints (4.1c), and Constraints (4.2f) to (4.2h) define the domain of decision variables. Since the scenarios are independent, we can determine the appropriate $M$ for each scenario $\omega \in \Omega$, represented by $M^\omega$, where $M^\omega = \max_{i \in I} \{T_i^\omega\} + \max_{i \in I, j \in I,\, i \neq j} \{S_{ij}\} (|I| - 1)$.

## 4.3 ETDRO approach for ASSP

In the ETO approach for ASSP, the probability distribution of uncertain parameters provided by the RF method is assumed to be the true distributional information and is directly used in the SP model. However, ML methods inherently produce some prediction errors, which cannot be entirely eliminated. Ignoring these potential prediction errors may result in unsatisfactory decisions. Therefore, we propose the ETDRO approach for ASSP. In this approach, the true probability distribution is assumed to be either completely unknown or only partially known, while the estimated probability distribution provided by the RF method is used as a known reference probability distribution in the DRO model.

In the ETDRO method, the use of the RF method to estimate the unknown param-

eter distribution is consistent with the ETO method. The main difference is that the ETDRO method employs the DRO model instead of the SP model for aircraft sequencing and scheduling plans. In this section, we first develop the DRO model for ASSP in Subsection 4.3.1. We then propose exact and inexact decomposition methods to solve the DRO model for ASSP based on the DR L-shaped method proposed by Bansal et al. (2018). The exact decomposition methods are introduced in Subsection 4.3.2, and the inexact decomposition methods based on the CPS method are proposed in Subsection 4.3.3.

### 4.3.1 DRO model for ASSP

The SP approach assumes that the probability distributions of uncertain parameters are known exactly. However, in real-world operations, these distributions are often not fully known. Considering this, the DRO approach can be employed, where only partial information about the distributions is required. Unlike the SP models, which optimise the expected value of a given function under a predetermined probability distribution, the DRO models aim to optimise the expected value of a given function for the worst-case probability distribution within an ambiguity set (Mohajerin Esfahani and Kuhn, 2018; Bansal et al., 2018; Agra and Rodrigues, 2022; Shehadeh, 2023; Tsang et al., 2024). The performance of DRO solutions is heavily dependent on the choice of ambiguity sets. In this study, we employ a type-1 Wasserstein ambiguity set, which is well suited to the scenarios generated by RF methods, given that a limited set of scenario data is considered in their construction. Besides, the type-1 Wasserstein ambiguity set offers the following advantages: it allows decision-makers to control their aversion to ambiguity by adjusting the radius of the Wasserstein ball and exhibit good convergence properties (Mohajerin Esfahani and Kuhn, 2018; Bansal et al., 2018; Agra and Rodrigues, 2022; Shehadeh, 2023).

In the DRO model for ASSP, the arrival and departure time for each aircraft $i \in I$

is modelled as a random variable with a probability distribution $P$ within a finite support $\Omega$. Since the DRO approach assumes that the true probability distribution $P$ is either completely unknown or only partially known, we consider the probability $\rho_P^\omega$ associated with each scenario $\omega \in \Omega$ belongs to the ambiguity set $\mathbb{P}$. The DRO model aims to minimise the expected makespan and aircraft delay costs under the worst-case probability distribution within the ambiguity set $\mathbb{P}$. We define the feasible region of first-stage variables $\boldsymbol{y}$ as $\mathcal{Y}$, and the formulation of the DRO model is written as follows:

$$\min_{\boldsymbol{y}\in\mathcal{Y}} \left\{ \max_{P\in\mathbb{P}} \mathbb{E}_{\mathbb{P}} \left[ Q\left(\boldsymbol{y}, \omega\right) \right] \right\} \Leftrightarrow \min_{\boldsymbol{y}\in\mathcal{Y}} \left\{ \max_{P\in\mathbb{P}} \sum_{\omega\in\Omega} \rho_P^\omega Q\left(\boldsymbol{y}, \omega\right) \right\}. \tag{4.3}$$

The DRO model for ASSP employs the type-1 Wasserstein ambiguity set to characterise ambiguity in the probability distribution. This ambiguity set defines an unknown probability distribution $P = \{\Omega, \rho_P\}$ that is close to a known reference probability distribution $P^* = \{\Omega, \rho_P^*\}$. More specifically, in the type-1 Wasserstein ambiguity set, $\Omega = \{1, 2, \ldots, |\Omega|\}$ represents a finite set of scenarios, $\rho_P := \left(\rho_P^1, \rho_P^2, \ldots, \rho_P^{|\Omega|}\right)$ denotes the probabilities related to an unknown reference probability distribution, and $\rho_{P*} := \left(\rho_{P*}^1, \rho_{P*}^2, \ldots, \rho_{P*}^{|\Omega|}\right)$ represents the given probabilities associated with a known reference probability distribution. Determining the type-1 Wasserstein distance between two distributions is viewed as a transportation problem for moving the probability mass from $\rho_{P^*}$ to $\rho_P$. The decision variables $m^{\omega\omega'}$ represent the probability mass moving from $\rho_P^\omega$ to $\rho_{P*}^{\omega'}$, and the parameter $\epsilon$ denotes the radius of the Wasserstein ball. Based on the above description and notations, for a given $\epsilon \geq 0$, the type-1 Wasserstein ambiguity set is formulated as follows:

$$\mathbb{W}_\epsilon = \left\{ \rho_P \in \mathbb{R}^{|\Omega|} : \sum_{\omega\in\Omega} \sum_{\omega'\in\Omega} \left\| \omega - \omega' \right\|_1 m^{\omega\omega'} \leq \epsilon \right. \tag{4.4a}$$

$$\sum_{\omega'\in\Omega} m^{\omega\omega'} = \rho_P^\omega, \quad \forall \omega \in \Omega, \tag{4.4b}$$

$$\sum_{\omega\in\Omega} m^{\omega\omega'} = \rho_{P*}^{\omega'}, \quad \forall\omega' \in \Omega, \tag{4.4c}$$

$$\sum_{\omega\in\Omega} \rho_P^{\omega} = 1, \tag{4.4d}$$

$$\rho_P^{\omega} \in \mathbb{R}^+, \quad \forall\omega \in \Omega, \tag{4.4e}$$

$$m^{\omega\omega'} \in \mathbb{R}^+, \quad \forall\omega \in \Omega, \forall\omega' \in \Omega \Big\}. \tag{4.4f}$$

### 4.3.2 Exact decomposition methods

In this subsection, the exact decomposition methods are proposed to solve the DRO model for ASSP. We first reformulate the DRO model (4.3) into the epigraph form, as shown in Model (4.5), where $\theta$ is the epigraphical decision variable.

$$\min \ \theta \tag{4.5a}$$

$$\text{s.t.} \quad \text{Constraints (4.1b), (4.1c),} \tag{4.5b}$$

$$\theta \geq \max\left\{\sum_{\omega\in\Omega} \rho_P^{\omega} Q\left(\boldsymbol{y}, \omega\right) : P \in \mathbb{P}\right\}, \tag{4.5c}$$

$$\theta \in \mathbb{R}^+. \tag{4.5d}$$

The DR L-shaped method proposed by Bansal et al. (2018), as a variant of the L-shaped method, is devised for solving the DRO model with the Wasserstein ambiguity set. When solving the DRO model using this method, three assumptions must be observed: (i) the feasible region $\mathcal{Y}$ defined by the first-stage variables $\boldsymbol{y}$ is nonempty; (ii) the model has relatively complete recourse property; and (iii) the scenario set $\Omega$ is finite. It is easy to find that ASSP satisfies assumptions (i) and (iii). However,

since the first-stage problem of the DRO model does not consider time constraints, an integer first-stage solution with errors in the aircraft landing and take-off sequence can be found within the feasible domain $\mathcal{Y}$, which may render the second-stage problems infeasible. This indicates that the ASSP does not satisfy the assumption (ii). Therefore, we should adapt the DR L-shaped method to the absence of the relatively complete recourse property. In addition to ensuring the applicability of the DR L-shaped method, we develop several algorithmic enhancements and inexact versions to improve its scalability.

Additionally, when using the DR L-shaped method for the DRO model of the ASSP, Constraint (4.5c) is replaced by a set of optimality cuts, thereby avoiding the explicit use of the second-stage problems $Q(\boldsymbol{y}, \omega)$. In Subsection 4.3.2.1, we introduce the DR L-shaped method, focusing on adapting it to solve the DRO model for ASSP without the relatively complete recourse property. In Subsection 4.3.2.2, we propose the E-DR L-shaped method, which incorporates time constraints, LBL cuts, and a warm start procedure to enhance computational performance. Finally, recognising the computational burden of solving the complex master problem at each iteration, we propose the E-DR-BBC method in Subsection 4.3.2.3. This method constructs a single branch-and-cut tree to solve the DRO model for ASSP.

#### 4.3.2.1 The DR L-shaped method

The DR L-shaped method proposed by Bansal et al. (2018) iterates between solving the master problem, the dual subproblems, and the distribution separation problem until the UB and the LB converge or the CPU time limit is reached. Each iteration begins with solving the master problem to obtain the first-stage solutions. Given these solutions, the dual subproblem for each scenario is solved to determine the optimal objective value of the corresponding second-stage problem. The distribution separation problem is then used to determine the probability of each scenario based on these

optimal objective values. Optimality cuts are generated from the solutions of the dual subproblems and the distribution separation problem, and these cuts are added to the master problem.

However, this basic framework cannot be directly applied to the DRO model for ASSP. Recall that the ASSP has no relatively complete recourse property, which may render the second-stage problems infeasible. Consequently, the corresponding dual subproblems may also be infeasible or unbounded, making it impossible to determine the optimal objective value of the second-stage problem. This issue prevents the generation of optimality cuts through the dual subproblems and the separation distribution problem, thus causing the convergence process of the DR L-shaped method to be interrupted (Bansal et al., 2018; Agra and Rodrigues, 2022). We introduce combinatorial cuts in the DR L-shaped method to address this issue. When an integer first-stage solution with errors in the aircraft landing and take-off sequence is identified, a combinatorial cut will be generated and added to the master problem. This ensures that the observed first-stage solutions with sequencing errors are excluded from the feasible domain $\mathcal{Y}$ in the next iteration. Conversely, suppose an integer first-stage solution without errors in the aircraft landing and take-off sequence is found. In that case, the dual subproblems and the distribution separation problem are solved sequentially to generate an optimality cut, which is then added to the master problem. The framework of the DR L-shaped method for the DRO model of the ASSP is presented in Algorithm 2.

The initial master problem of the DR L-shaped method is formulated as follows:

$$\min \quad \theta \tag{4.6a}$$

$$\text{s.t.} \quad \text{Constraints (4.1b), (4.1c), (4.5d).} \tag{4.6b}$$

After solving the master problem in an iteration, we obtain the aircraft sequencing

**Algorithm 2** The DR L-shaped method

**While** UB - LB $> 0$ or $CPU_{limit}$ is not reached **do**.
  Solve the master problem (4.6)
  **If** there are no subtours in $\hat{\mathbf{y}}$ **then**
    Solve subproblems (4.8) with $\hat{\mathbf{y}}$ for each scenario.
    Solve distribution separation problem (4.9).
    Add Benders cut (4.10) to the master problem (4.6).
  **Else**
    Add combinatorial cut (4.7) to the master problem (4.6).
  **End if**
**End while**

decisions $\hat{\boldsymbol{y}}$. The sequence position of aircraft $i$ is given by $\left(|I| - \sum_{j \in I \setminus \{i\}} \hat{y}_{ij}\right)$. If two aircraft have the same sequence position, an integer first-stage solution with errors in the aircraft landing and take-off sequence is found. To prevent the recurrence of first-stage solutions with sequencing errors in subsequent iterations, a combinatorial cut (4.7) is added to the master problem, $\mathcal{S} = \{(i,j) \,|\hat{y}_{ij} = 1,\ \forall i \in I,\ \forall j \in I,\ i \neq j\}$. Subsequently, the current iteration concludes, and the next iteration commences.

$$\sum_{(i,j) \in \mathcal{S}} y_{ij} \leq |\mathcal{S}| - 1. \tag{4.7}$$

If the aircraft sequencing decisions provided by the master problem are free of sequencing errors, we establish that LB $= \hat{\theta}$. Subsequently, we solve the dual subproblem for each scenario. The primal subproblem for scenario $\omega \in \Omega$ is formulated as Model (4.2). By introducing dual variables $\pi_i^\omega$, $\tau_{ij}^\omega$, $\delta_i^\omega$, and $\sigma_i^\omega$ for Constraints (4.2b) to (4.2e), the dual subproblem under scenario $\omega$ is written as follows:

$$\max \quad \sum_{i \in I} T_i^\omega \pi_i^\omega + \sum_{i \in I} \sum_{j \in I \setminus \{i\}} \left(M\left(1 - \hat{y}_{ij}\right) - S_{ij}\right) \tau_{ij}^\omega + A^\omega \sum_{i \in I} \delta_i^\omega + \sum_{i \in I} T_i^\omega \sigma_i^\omega \tag{4.8a}$$

$$\text{s.t.} \quad \pi_i^\omega + \sum_{j \in I \setminus \{i\}} \tau_{ij}^\omega - \sum_{j \in I \setminus \{i\}} \tau_{ji}^\omega + \delta_i^\omega + \sigma_i^\omega \leq 0, \quad \forall i \in I, \tag{4.8b}$$

$$-\sum_{i\in I}\delta_i^{\omega} \leq W_1, \tag{4.8c}$$

$$-\sigma_i^{\omega} \leq W_2 C_i^{\text{delay}}, \quad \forall i \in I, \tag{4.8d}$$

$$\pi_i^{\omega} \in \mathbb{R}^{+}, \quad \forall i \in I, \tag{4.8e}$$

$$\tau_{ij}^{\omega} \in \mathbb{R}^{-}, \quad \forall i \in I,\ \forall i \in I,\ i \neq j, \tag{4.8f}$$

$$\delta_i^{\omega} \in \mathbb{R}^{-}, \quad \forall i \in I, \tag{4.8g}$$

$$\sigma_i^{\omega} \in \mathbb{R}^{-}, \quad \forall i \in I. \tag{4.8h}$$

Recall that we do not impose a UB on the scheduled time $t_i^{\omega}$ for each aircraft $i$ under each scenario $\omega$. Consequently, all first-stage solutions free of sequencing errors are feasible for the second-stage problems, and optimal objective value $\hat{Q}\left(\boldsymbol{y}, \omega\right)$ for the second-stage problem under each scenario can be found. Based on these optimal objective values, we solve the distribution separation problem (4.9) corresponding to the Wasserstein ambiguity set.

$$\max\left\{\sum_{\omega\in\Omega}\rho_P^{\omega}\hat{Q}\left(\boldsymbol{y}, \omega\right) \mid \rho_P \in \mathbb{W}_{\epsilon}\right\}. \tag{4.9}$$

After solving the distribution separation problem, we determine the UB of the method as $\text{UB} = \min\left\{\text{UB}, \sum_{\omega\in\Omega}\hat{\rho}_P^{\omega}\hat{Q}\left(\boldsymbol{y}, \omega\right)\right\}$. When the UB converges with the LB, the DR L-shaped method terminates. Otherwise, we utilise the probability $\hat{\rho}_P^{\omega}$ for each scenario $\omega$, as provided by the distribution separation problem, along with the optimal solutions of the dual subproblems $\hat{\pi}_i^{\omega}$, $\hat{\tau}_{ij}^{\omega}$, $\hat{\delta}_i^{\omega}$, and $\hat{\sigma}_i^{\omega}$, to generate the optimality cut (4.10) and incorporate it into the master problem. We then commence the next iteration.

**Algorithm 3** The E-DR L-shaped method

**While** UB - LB $> 0$ or $CPU_{limit}$ is not reached **do**.
  Solve the master problem (4.11)
  **If** there are no subtours in $\hat{\mathbf{y}}$ **then**
    Solve subproblems (4.8) with $\hat{\mathbf{y}}$ for each scenario.
    Solve distribution separation problem (4.9).
    Add Benders cut (4.10) to the master problem (4.11).
  **Else**
    Add combinatorial cut (4.7) to the master problem (4.11).
  **End if**
**End while**

$$\theta \geq \sum_{\omega \in \Omega} \hat{\rho}_P^{\omega} \left( \sum_{i \in I} T_i^{\omega} \hat{\pi}_i^{\omega} + \sum_{i \in I} \sum_{j \in I \setminus \{i\}} (M - S_{ij}) \hat{\tau}_{ij}^{\omega} - M \sum_{i \in I} \sum_{j \in I \setminus \{i\}} \hat{\tau}_{ij}^{\omega} y_{ij} + A^{\omega} \sum_{i \in I} \hat{\delta}_i^{\omega} + \sum_{i \in I} T_i^{\omega} \hat{\sigma}_i^{\omega} \right). \tag{4.10}$$

#### 4.3.2.2 The E-DR L-shaped method

In Subsection 4.3.2.1, we focus on adapting the DR L-shaped method to solve ASSP that lacks the relatively complete recourse property. In this subsection, we further propose an E-DR L-shaped method, where some algorithmic enhancements are included to improve its computational efficiency. The primary distinction between the E-DR L-shaped and DR L-shaped methods lies in the master problem. In the E-DR L-shaped method, time constraints and LBL cuts are incorporated into the master problem, along with the initial optimality cuts provided by the warm start procedure. The framework of the E-DR L-shaped method for the DRO model of the ASSP is presented in Algorithm 3.

The initial master problem of the E-DR L-shaped method is formulated as follows:

$$\min \quad \theta \tag{4.11a}$$

$$\text{s.t.} \quad \text{Constraints (4.1b), (4.1c), (4.5d),} \tag{4.11b}$$

$$t_i^\omega + S_{ij} - t_j^\omega \leq M(1 - y_{ij}), \quad \forall i \in I,\ \forall j \in I,\ i \neq j,\ \forall \omega \in \Omega, \tag{4.11c}$$

$$z^\omega \geq t_i^\omega - A^\omega, \quad \forall i \in I,\ \forall \omega \in \Omega, \tag{4.11d}$$

$$d_i^\omega \geq t_i^\omega - T_i^\omega, \quad \forall i \in I,\ \forall \omega \in \Omega, \tag{4.11e}$$

$$\theta \geq \sum_{\omega \in \Omega} \rho_{P*}^\omega \left( W_1 z^\omega + W_2 \sum_{i \in I} C_i^{\text{delay}} d_i^\omega \right), \tag{4.11f}$$

$$\begin{aligned} \theta \geq \sum_{\omega \in \Omega} \hat{\rho}_P^{\omega\zeta} \Bigg( & \sum_{i \in I} T_i^\omega \hat{\pi}_i^{\omega\zeta} + \sum_{i \in I} \sum_{j \in I \setminus \{i\}} (M - S_{ij}) \hat{\tau}_{ij}^{\omega\zeta} \\ & - M \sum_{i \in I} \sum_{j \in I \setminus \{i\}} \hat{\tau}_{ij}^{\omega\zeta} y_{ij} + A^\omega \sum_{i \in I} \hat{\delta}_i^{\omega\zeta} + \sum_{i \in I} T_i^\omega \hat{\sigma}_i^{\omega\zeta} \Bigg), \quad \forall \zeta \in Z, \end{aligned} \tag{4.11g}$$

$$t_i^\omega \in \mathbb{R}^+, \quad \forall i \in I,\ \forall \omega \in \Omega, \tag{4.11h}$$

$$z^\omega \in \mathbb{R}^+, \quad \forall \omega \in \Omega, \tag{4.11i}$$

$$d_i^\omega \in \mathbb{R}^+, \quad \forall i \in I,\ \forall \omega \in \Omega, \tag{4.11j}$$

In the DR L-shaped method, combinatorial cuts are used to eliminate solutions with sequencing errors, but they do not directly improve the LB. Cuts that fail to enhance the LB are generally considered undesirable (Rahmaniani, Ragheb and Crainic, Teodor Gabriel and Gendreau, Michel and Rei, Walter, 2017; Ng et al., 2021). An alternative

approach is to incorporate time constraints (4.11c) from the second-stage problem of each scenario into the master problem, thereby excluding solutions with sequencing errors (Agra and Rodrigues, 2022; Wang et al., 2022b).

Since part of the original objective function is projected out in the master problem of the BD method (also known as the L-shaped method), the initial optimality gap can be substantial (Rahmaniani, Ragheb and Crainic, Teodor Gabriel and Gendreau, Michel and Rei, Walter, 2017; Rahmaniani et al., 2018; Adulyasak et al., 2015; Wu et al., 2022; Shehadeh, 2023; Tsang et al., 2024). To narrow this optimality gap, the BD method requires numerous iterations and the generation of multiple optimality cuts. The LBL cut, which incorporates information from the excluded part of the original objective function, can be added to the master problem to improve the LB of the BD method (Adulyasak et al., 2015; Wu et al., 2022; Shehadeh, 2023; Tsang et al., 2024). Inspired by this approach, we design the LBL cut (4.11f) to enhance the LB of the DR L-shaped method. The makespan and delay times under each scenario in the LBL cut (4.11f) are determined by Constraints (4.11d) and (4.11e), respectively. To demonstrate the validity of the proposed LBL cut (4.11f), we begin with the following proposition.

**Proposition 4.3.1.** *The LBL cut (4.11f) is a valid cut.*

*Proof*: Let $\boldsymbol{y}$ denote a feasible first-stage solution without sequencing errors, and let the optimal objective value for the second-stage problem under each scenario be denoted as $Q(\boldsymbol{y}, \omega)$. The distribution separation problem with the type-1 Wasserstein ambiguity set seeks to identify the unknown probability distribution $\rho_P^\omega$ close to the reference probability distribution $\rho_{P*}^\omega$ for each scenario $\omega$ that maximises the inner optimisation problem in Model (4.3). Consequently, we have:

$$\max \left\{ \sum_{\omega \in \Omega} \rho_P^\omega Q(\boldsymbol{y}, \omega) \mid \rho_P \in \mathbb{W}_\epsilon \right\} \geq \sum_{\omega \in \Omega} \rho_{P*}^\omega Q(\boldsymbol{y}, \omega) \tag{4.12}$$

Based on the Objective function (4.2a) and the Constraints (4.5c), we have:

$$\sum_{\omega\in\Omega}\rho^{\omega}_{P*}Q(\boldsymbol{y},\omega)\geq\sum_{\omega\in\Omega}\rho^{\omega}_{P*}\left(W_1z^{\omega}+W_2\sum_{i\in I}C_i^{\text{delay}}d_i^{\omega}\right) \tag{4.13}$$

and

$$\theta\geq\max\left\{\sum_{\omega\in\Omega}\rho^{\omega}_{P}Q(\boldsymbol{y},\omega)\mid\rho_P\in\mathbb{W}_{\epsilon}\right\} \tag{4.14}$$

Thus, we can obtain:

$$\begin{aligned}\theta&\geq\max\left\{\sum_{\omega\in\Omega}\rho^{\omega}_{P}Q(\boldsymbol{y},\omega)\ \mid\ \rho_P\in\mathbb{W}_{\epsilon}\right\}\\&\geq\sum_{\omega\in\Omega}\rho^{\omega}_{P*}Q(\boldsymbol{y},\omega)\\&\geq\sum_{\omega\in\Omega}\rho^{\omega}_{P*}\left(W_1z^{\omega}+W_2\sum_{i\in I}C_i^{\text{delay}}d_i^{\omega}\right)\end{aligned} \tag{4.15}$$

That is the LBL cut (4.11f). Moreover, the LBL cut (4.11f) has no restrictions on the feasible domain of the master problem. Therefore, the LBL cut (4.11f) is valid. □

Besides, the warm start procedure generates several initial cuts (4.11g). Initially, we solve the SP model (4.1) using the same inputs as the DRO model (4.3) to obtain the optimal first-stage sequencing decisions, denoted as $\boldsymbol{y}^{\text{SP}}$. Subsequently, leveraging the CPS method (Balakrishnan and Chandran, 2010), where $k=1$, we generate a set of sequencing decisions in which each aircraft's position remains closely aligned with its position in the original sequencing decisions $\boldsymbol{y}^{\text{SP}}$. For those generated sequencing decisions, we solve the dual subproblems (4.8) and the distribution separation problem (4.9) to obtain their objective values and associated optimality cuts. Optimality

**Algorithm 4** The E-DR-BBC method

Solve the master problem (4.11) with a $CPU_{limit}$.
**If** an integer first-stage solution $\hat{\mathbf{y}}$ is found **then**
    **If** there are no subtours in $\hat{\mathbf{y}}$ **then**
        Solve subproblems (4.8) with $\hat{\mathbf{y}}$.
        Solve distribution separation problem (4.9).
        Add Benders cuts (4.10) to the unfathomed nodes.
    **Else**
        Add subtour elimination cuts (4.7) to the unfathomed nodes.
    **End if**
**End if**

cuts derived from sequencing decisions generated by the CPS method, with objective values less than or equal to $f^{\text{DRO}}(\boldsymbol{y}^{\text{SP}})$, are classified as strong optimality cuts and incorporated into the initial cut set $Z = \{1, 2, \ldots, |Z|\}$.

#### 4.3.2.3 The E-DR-BBC method

Solving the master problems of the DR L-shaped and E-DR L-shaped methods for the DRO model of the ASSP is time-consuming due to their mixed integer programming nature, and the addition of cuts in each iteration can further increase their complexity (Rei et al., 2009). Instead of solving the mixed integer linear programming master problem in each iteration, the BBC method constructs a single branch-and-cut tree, where cuts are added to the unfathomed nodes after an integer first-stage solution is found in the branch-and-cut search (Gendron et al., 2016). Based on the BBC method, we proposed the E-DR-BBC method for the DRO model of the ASSP. The framework of the E-DR-BBC method is presented in Algorithm 4.

### 4.3.3 Inexact decomposition method

The ASSP is typically considered a near-real-time optimisation problem that requires rapid decision-making to ensure efficient and safe runway operations. However, the

ASSP exhibits an NP-hard nature. The proposed DRO model for ASSP further accounts for the uncertainty and ambiguity of aircraft arrival and departure times, making it challenging to achieve an optimal solution. The exact decomposition methods discussed in Subsection 4.3.2 require a long CPU time. To enhance the scalability of the DRO model for ASSP, we propose the inexact decomposition method based on the CPS method.

The primary distinction between the inexact decomposition method and the exact decomposition method lies in the stabilisation of the master problem around a stability centre point. This stabilisation reduces the feasible region and facilitates the generation of strong Benders cuts. Consequently, this proposed inexact decomposition method is termed the S-DR-BBC method. However, it should be noted that the optimal solution of the stabilised master problem does not provide a valid global lower bound, as it only considers a portion of the original problem's feasible region. The initial stabilised master problem of the S-DR-BBC methods is formulated as follows:

$$\min \quad \theta \tag{4.16a}$$

$$\text{s.t.} \quad \text{Constraints (4.1b), (4.1c), (4.5d), (4.11c)-(4.11j),} \tag{4.16b}$$

$$\max\{1,(l_i-k)\} \leq \left(|I| - \sum_{j\in I\setminus\{i\}} y_{ij}\right) \leq \min\{|I|,(l_i+k)\}, \quad \forall i \in I, \tag{4.16c}$$

The stabilised master problem (4.16) explores new solutions in the vicinity of a stability centre point (i.e., a feasible first-stage solution) through the CPS constraint (4.16c). Specifically, an aircraft cannot be shifted by more than $k$ positions from its position in a feasible sequence. Incorporating the CPS constraint (4.16c) makes the stabilised master problem (4.16) easier to solve by reducing its feasible region. However, it should be noted that the optimal solution for the S-DR-BBC method

does not provide a valid global LB because it is only considered part of the feasible region of the original problem.

## 4.4 Numerical experiments for ASSP

### 4.4.1 Experimental design for ASSP

This subsection designs the numerical experiments using data from XMN. As illustrated in Figure 4.1, XMN currently features a runway (05/23) measuring 3,400 meters in length and 45 meters in width, and is one of the busiest single-runway airports in China.

The test instances are generated using data from 31st October 2023. The number of arriving and departing aircraft at hourly intervals is shown in Figure 4.2. Given that aircraft sequencing and scheduling decisions have a near real-time characteristic, we use a 20-minute decision interval to segment the day's data, as recommended by Solak et al. (2018). After removing test instances with one or fewer aircraft, the total number of test instances is 56. The details of each test instance are provided in Table 4.5. The ID of a test instance consists of two numbers. The first number denotes the hour of the instance, while the second number indicates the specific 20-minute period within that hour. For example, the data set from 09:00 to 09:20 is represented as 9_1, and the data set from 22:40 to 23:00 is represented as 22_3. We adopt the SAA method as the scenario selection strategy for each test instance, with the number of scenarios set to $|\Omega| = 100$.

The aircraft arriving and departing XMN are primarily heavy and large types. Table 4.6 presents the separation time requirements for aircraft type combinations used in this study, following the settings of Pohl et al. (2021, 2022). To account for the greater importance of delays affecting larger aircraft with more passengers (Pohl et al., 2021), we use delay cost coefficients $C_i^{\text{delay}}$ of 1 and 2 monetary units for large


Figure 4.1: Runway (05/23) at XMN Airport.

and heavy aircraft types, respectively. For the numerical experiments, we set $W_1$ and $W_2$ to 0.5.

### 4.4.2 An illustrative example for ASSP

In this subsection, we comprehensively analyse test instance 8_1. Table 4.7 presents detailed information, including the estimated arrival and departure times $T_i$, the aircraft's landing or take-off status, and the aircraft types.

In Figure 4.3, we provide graphical representations of the aircraft sequencing and

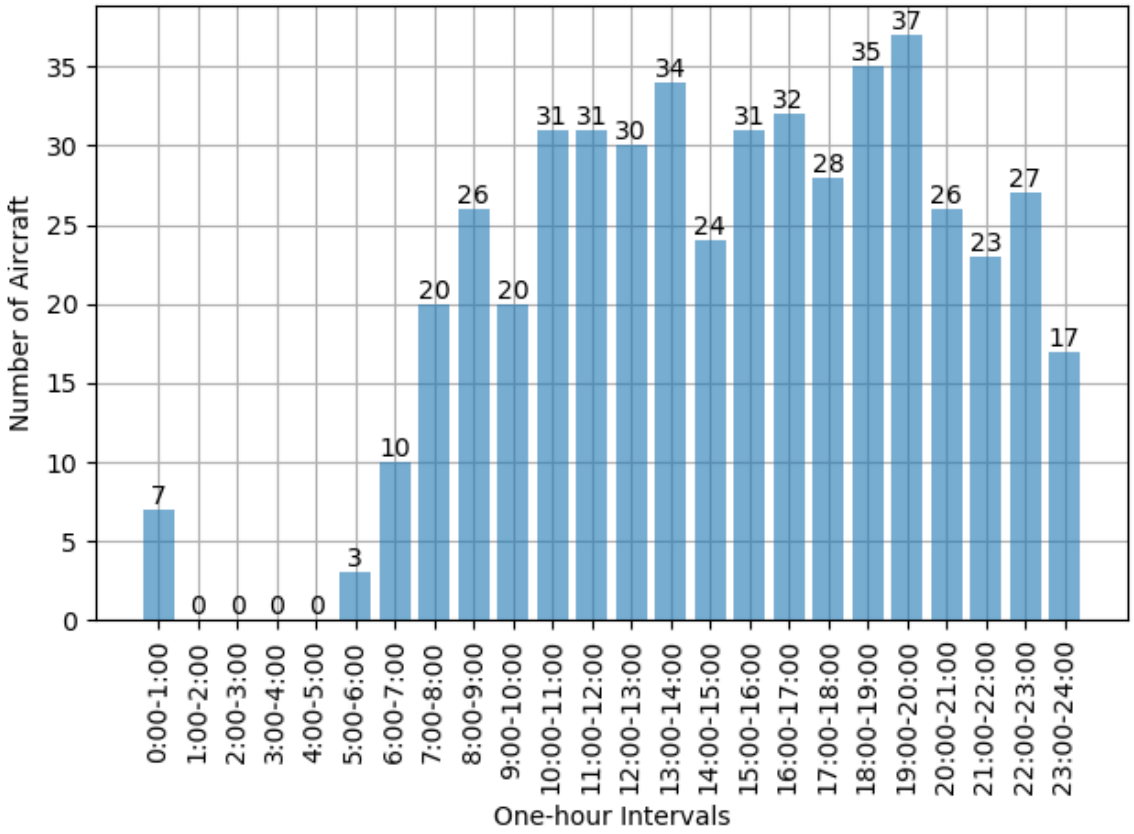


Figure 4.2: Number of arriving and departing aircraft at hourly intervals on 31st October 2023 at XMN, China.

scheduling plans generated by the deterministic model, the ETO approach, and the ETDRO approach. These plans may exhibit structural differences because each approach is designed to optimise under specific conditions. The ETO approach is designed to optimise an expected function based on a specific reference probability distribution provided by the ML method. In contrast, the ETDRO approach accounts for potential deviations in the distribution of uncertain parameters from the reference distribution given by the ML method. Consequently, it optimises the expected value of a function under the worst-case probability distribution defined by the Wasserstein ambiguity set. It should be noted that the ETO approach is equivalent to the ETDRO approach when

Table 4.5: Detailed information of test instances for ASSP.

| Test instances | Aicraft number | Test instances | Aicraft number |
|---|---|---|---|
| 0_1 | 5 | 14_3 | 8 |
| 0_2 | 2 | 15_1 | 7 |
| 5_2 | 3 | 15_2 | 14 |
| 6_2 | 2 | 15_3 | 10 |
| 6_3 | 8 | 16_1 | 9 |
| 7_1 | 5 | 16_2 | 12 |
| 7_2 | 8 | 16_3 | 11 |
| 7_3 | 7 | 17_1 | 10 |
| 8_1 | 10 | 17_2 | 11 |
| 8_2 | 11 | 17_3 | 6 |
| 8_3 | 5 | 18_1 | 11 |
| 9_1 | 4 | 18_2 | 13 |
| 9_2 | 9 | 18_3 | 11 |
| 9_3 | 7 | 19_1 | 12 |
| 10_1 | 9 | 19_2 | 13 |
| 10_2 | 10 | 19_3 | 12 |
| 10_3 | 12 | 20_1 | 6 |
| 11_1 | 9 | 20_2 | 11 |
| 11_2 | 12 | 20_3 | 9 |
| 11_3 | 10 | 21_1 | 11 |
| 12_1 | 10 | 21_2 | 7 |
| 12_2 | 10 | 21_3 | 5 |
| 12_3 | 10 | 22_1 | 7 |
| 13_1 | 13 | 22_2 | 10 |
| 13_2 | 13 | 22_3 | 10 |
| 13_3 | 8 | 23_1 | 6 |
| 14_1 | 8 | 23_2 | 6 |
| 14_2 | 8 | 23_3 | 5 |

$\epsilon = 0$. Given the inherent uncertainty in aircraft arrival and departure times, which cannot be precisely known in advance, we use nominal times for each aircraft as input to the deterministic model. This model can be viewed as an SP model with a single scenario consisting of these nominal times. The ETDRO approach with a given $\epsilon$ is denoted as ETDRO$_\epsilon$. As shown in Figure 4.3, the plan provided by the deterministic

Table 4.6: Separation time requirements for landing and take-off aircraft (in seconds)

| Leading | Trailing | | | |
|---|---|---|---|---|
| | Landing | | Take-off | |
| | Heavy | Large | Heavy | Large |
| Landing | | | | |
| Heavy | 96 | 157 | 75 | 75 |
| Large | 60 | 69 | 75 | 75 |
| Take-off | | | | |
| Heavy | 60 | 60 | 90 | 120 |
| Large | 60 | 60 | 60 | 60 |

Table 4.7: Details of the test instance 8_1 for ASSP.

| Aircraft | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $T_i$ | 29,700 | 30,000 | 29,700 | 29,400 | 30,000 | 30,600 | 30,000 | 27,900 | 30,600 | 31,200 |
| LD/TO | TO | TO | TO | TO | TO | TO | LD | LD | TO | LD |
| Aircraft type | LG | LG | LG | LG | LG | HV | LG | HV | LG | LG |

Note: 'LD' indicates landing, 'TO' indicates take-off, 'HV' indicates heavy, and 'LG' indicates large.

model is structurally very different from those provided by the ETO and ETDRO approaches. The structural differences between the ETO and ETDRO plans are relatively minor. However, these differences become more pronounced as the $\epsilon$ value increases.

Next, we present performance indicators for these optimisation approaches in Table 4.8. Our findings indicate that the objective values from the deterministic model are significantly smaller than those from both the ETO and ETDRO approaches. This difference arises because the deterministic model optimises for a single, specific scenario. In contrast, the ETO and ETDRO approaches must optimise across multiple possible scenarios to ensure performance in each scenario. The complexity of optimising for various scenarios results in higher objective values. Moreover, the parameter $\epsilon$ used in the ETDRO approach significantly impacts the objective values. As the value

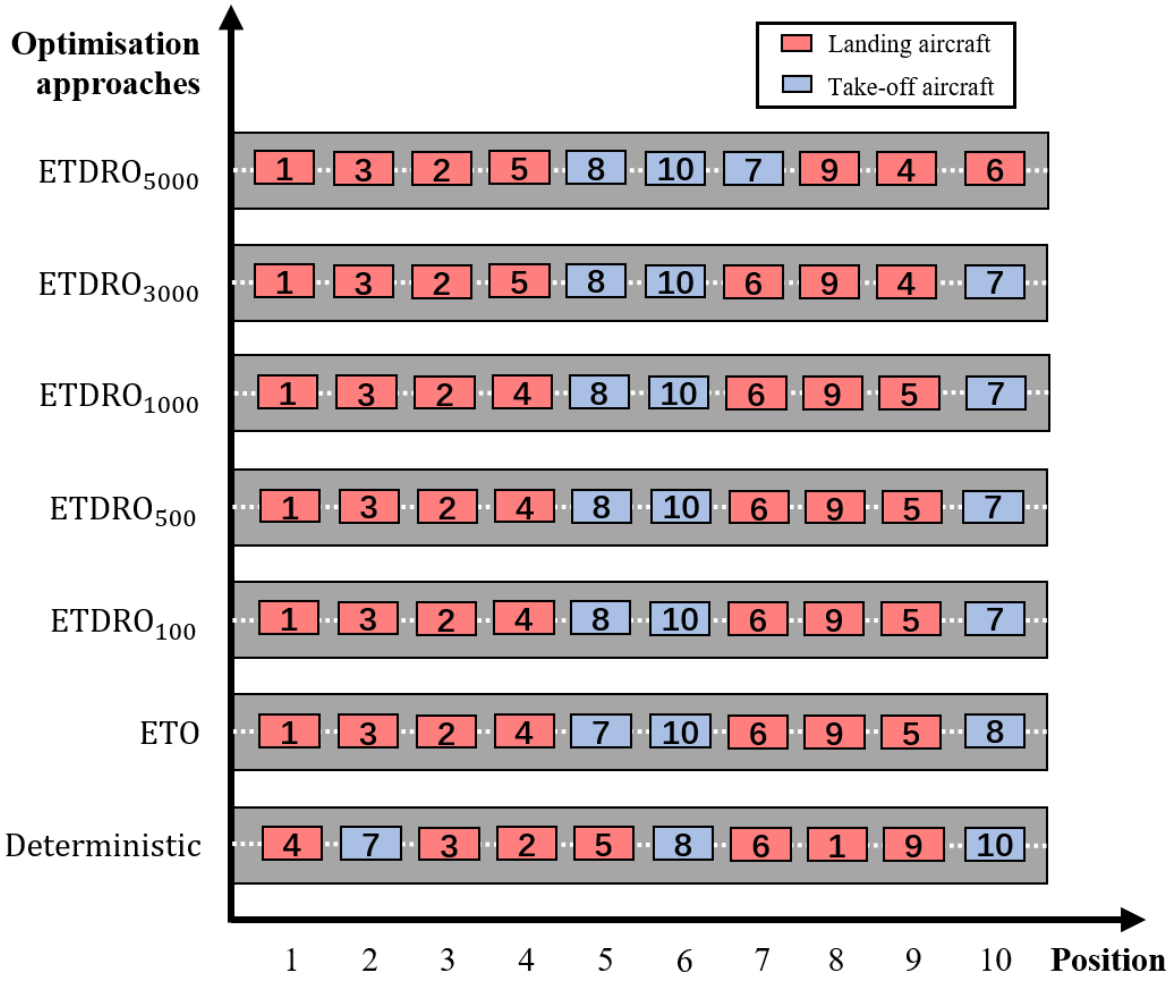


Figure 4.3: Aircraft sequencing and scheduling plans provided by optimisation approaches for test instance 8_1.

of $\epsilon$ increases, the objective value of the solutions also increases. Table 4.8 also shows that the mean value of the ETDRO approach is higher than that of the ETO approach. However, the ETDRO approach outperforms the ETO approach in extreme situations within the in-sample data, as evidenced by the smaller 95-quantile and 99-quantile values. These observations suggest that the ETDRO approach yields more conservative aircraft sequencing and scheduling plans when faced with in-sample data. While this results in higher mean values, the approach performs better under extreme conditions.

Table 4.8: Evaluation of the obtained aircraft sequencing and scheduling plans for test instance 8_1 for ASSP.

| | Deterministic | ETO | $\text{ETDRO}_{100}$ | $\text{ETDRO}_{500}$ | $\text{ETDRO}_{1000}$ | $\text{ETDRO}_{3000}$ | $\text{ETDRO}_{5000}$ |
|---|---|---|---|---|---|---|---|
| Objective value | **1,822.50** | 9,382.52 | 9,571.01 | 10,250.68 | 11,000.59 | 13,546.56 | 15,859.94 |
| Mean | 16,597.73 | **9,382.52** | 9,389.22 | 9,389.22 | 9,389.22 | 9,432.55 | 9,482.76 |
| 0.95-quantile | 59,008.70 | 21,299.00 | 21,297.57 | 21,297.57 | 21,297.57 | 20,730.65 | **18,419.00** |
| 0.99-quantile | 12,0629.28 | 34,742.23 | **33,462.44** | **33,462.44** | **33,462.44** | 33,838.64 | 34,114.54 |

### 4.4.3 In-sample analysis for ASSP

In this subsection, we perform an in-sample analysis using the scenario set $\Omega$ of size 100 to evaluate the performance of the ETO and ETDRO approaches. We first utilise the value of the stochastic solution (VSS) indicator to evaluate whether it is worth modelling uncertain aircraft arrival and departure times in the ASSP using the ETO approach. The VSS is defined as the difference between the expected value of the expected value solution (EEV) and the optimal objective value of the SP model (Birge and Louveaux, 2011). In this study, the SP model is replaced by the ETO approach. The formulation of VSS is presented as follows:

$$\text{VSS} = \text{EEV} - \text{ETO}, \tag{4.17}$$

and the relative VSS is calculated as the ratio:

$$\text{VSS}\ (\%) = \frac{(\text{EEV} - \text{ETO})}{\text{ETO}} * 100. \tag{4.18}$$

Table 4.9 shows the average results of 56 test instances related to the value of incorporating stochasticity. The average value of the relative VSS is 22.06%. The results indicate that the aircraft sequencing and scheduling plans provided by the deterministic ASSP model can incur average additional costs of 22.06% compared to those provided by the ETO approach of the ASSP.

Table 4.9: Value of incorporating stochasticity for ASSP

| | EEV | ETO | VSS | VSS(%) |
|---|---|---|---|---|
| Average | 6949.28 | 5693.53 | 1255.76 | 22.06 |

Table 4.10: Value of using the DRO model for ASSP

| | Deterministic | ETO | $\text{ETDRO}_{100}$ | $\text{ETDRO}_{500}$ | $\text{ETDRO}_{1000}$ | $\text{ETDRO}_{3000}$ | $\text{ETDRO}_{5000}$ |
|---|---|---|---|---|---|---|---|
| Mean | 6,974.73 | **5,695.37** | 5,696.68 | 5,724.71 | 5,749.38 | 5,897.62 | 5,974.97 |
| 0.95-quantile | 15,867.26 | 11,872.51 | 11,863.05 | 11,845.95 | 11,830.35 | 11,874.27 | **11,826.88** |
| 0.99-quantile | 24,123.03 | 18,141.71 | 18,018.21 | 17,597.03 | 17,459.39 | 17,169.71 | **16,901.51** |
| Worst-case | 28,721.72 | 22,185.38 | 21,926.96 | 20,917.32 | 20,621.54 | 19,759.36 | **19,373.84** |

We then evaluate the value of using the ETDRO approach, considering different radii of the Wasserstein ball, i.e., $\epsilon = \{100, 500, 1000, 3000, 5000\}$. For each first-stage solution provided by the models, we solve the corresponding second-stage problem (4.2) to obtain the second-stage costs for each in-sample scenario. Since our ASSP model has no objective function in the first stage, the second-stage costs represent the total costs. Table 4.10 presents performance statistics of the in-sample total costs obtained by different models. We find that the performance statistics of both the ETO and ETDRO approaches are much better than those of the deterministic model. In the in-sample analysis, we found that the ETO approach performed better for the average indicator, while the ETDRO approach excelled in the 0.95-quantile, 0.99-quantile, and worst-case scenario indicators. The ETO approach performs better on the average indicator, suggesting it may be more cost-efficient in relatively stable and predictable operating environments. Runway controllers can leverage the ETO approach for routine decision-making in such conditions. Conversely, the ETDRO approaches' superior performance in the 0.95-quantile, 0.99-quantile, and worst-case indicators demonstrate their robustness under high uncertainty and extreme conditions. Runway controllers should consider using the ETDRO approach in volatile operating environments or when the costs of extreme events are significant.

### 4.4.4 Out-of-sample analysis for ASSP

In this subsection, we conduct out-of-sample analysis to assess the performance of models on unseen data, evaluating their generalisation ability and predictive performance. This is crucial for verifying whether the models can effectively handle new data and uncertainty in real-world applications. First, the optimal solution $\hat{\mathbf{y}}^*$ provided by each optimisation approach is fixed. Then, we solve the second-stage problem $\mathbb{E}_{\omega \in \Omega_{\text{out}}} [Q(\hat{\mathbf{y}}^*, \omega)]$ with an out-of-sample scenario set $\Omega_{\text{out}}$ of size 1000, which is generated in the same way as the scenario set $\Omega$.

Table 4.11 presents the performance statistics of the out-of-sample total costs obtained by different models under various scenario set sizes. Consistent with the in-sample analysis, the performance statistics of both the ETO and ETDRO approaches are significantly superior to those of the deterministic model in the out-of-sample analysis. For the average performance, the $\text{ETDRO}_{100}$ approach demonstrates superior performance with a value of 5,880.45, outperforming the deterministic model and the ETO approach. As the radius of the Wasserstein sphere increases, the average performance of the ETDRO approach generally declines, suggesting that larger radii result in diminishing returns in average performance when dealing with unknown data. For the 0.95- and 0.99-quantile indicators, the $\text{ETDRO}_{100}$ approach consistently outperforms both the deterministic model and ETO approach, particularly at higher quantiles. The $\text{ETDRO}_{1000}$ approach performs best at the 0.99-quantile, demonstrating the robustness of the approach at the tail end of the distribution. Additionally, in the worst-case scenario, the $\text{ETDRO}_{3000}$ approach achieves the lowest value, indicating its superior effectiveness in managing extreme situations compared to other approaches. Adopting the ETDRO approach enhances average and quantile performance, indicating that decision-makers should consider implementing the ETDRO approach, particularly $\text{ETDRO}_{100}$, to achieve more reliable and efficient outcomes. Overall, based on the results

Table 4.11: Results of out-of-sample analysis for ASSP

| | Deterministic | ETO | $\text{ETDRO}_{100}$ | $\text{ETDRO}_{500}$ | $\text{ETDRO}_{1000}$ | $\text{ETDRO}_{3000}$ | $\text{ETDRO}_{5000}$ |
|---|---|---|---|---|---|---|---|
| Mean | 6,992.72 | 5,882.31 | **5,880.45** | 5,908.68 | 5,947.99 | 6,097.33 | 6,169.61 |
| 0.95-quantile | 16,360.67 | 12,475.50 | 12,448.61 | **12,402.57** | 12,411.57 | 12,437.43 | 12,414.19 |
| 0.99-quantile | 25,413.65 | 20,670.53 | 20,496.89 | 20,281.20 | 20,232.32 | 20,249.17 | **20,158.11** |
| Worst-case | 40,034.06 | 34,664.70 | 34,495.93 | 34,080.52 | 33,896.96 | 33,567.15 | **33,306.57** |

of in-sample and out-of-sample analysis, the $\text{ETDRO}_{100}$ approach exhibits relatively low values in terms of average, 0.95-quantile, 0.99-quantile, and worst-case indicators, demonstrating its effectiveness in both normal and extreme scenarios.

### 4.4.5 Actual sample analysis for ASSP

In this subsection, we compare the effectiveness of aircraft sequencing and scheduling plans provided by different approaches in actual scenarios. To achieve this, we employ a method that integrates actual scenarios called actual sample analysis. First, we fix the optimal solution $\hat{\mathbf{y}}^*$ provided by each model. Then, we solve the second-stage problem $Q\left(\hat{\mathbf{y}}^*, \omega^{\text{actual}}\right)$, where $\omega^{\text{actual}}$ represents the actual scenario. Considering that in previous ASSP studies, the analysis based on historical data has been the primary source for the distribution of uncertain parameters (Solveling et al., 2011; Solak et al., 2018). Therefore, the actual sample analysis not only compares the ETO and ETDRO approaches with the deterministic model but also includes comparisons with the SP and DRO models based on the probability distribution of aircraft arrival and departure times derived from historical data. We present the probability distribution of aircraft arrival and departure time deviation based on historical data from 1st September to 30th October 2023 in Figure 4.4, where outliers are eliminated using the interquartile range (IQR) method. Subsequently, scenarios $\omega \in \Omega$ can be constructed by generating deviations $\hat{a}_i$ in aircraft arrival or departure times from the estimated time $T_i$, where $\hat{a}_i$ is randomly generated from the probability distribution of aircraft arrival or departure time deviation shown in Figure 4.4(a) and Figure 4.4(b), respectively.

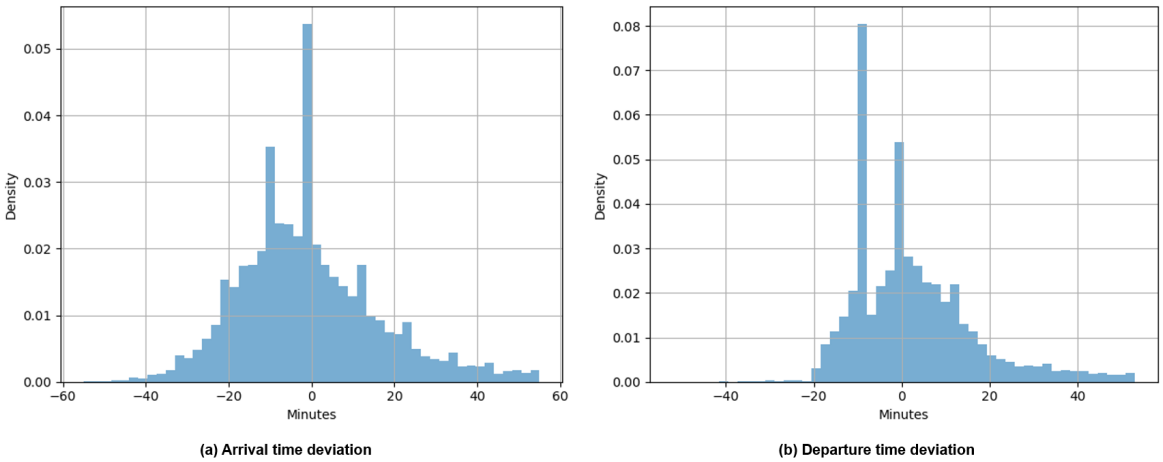


Figure 4.4: Distribution of aircraft arrival and departure time deviation based on historical data at XMN.

The overall results presented in Figure 4.5 illustrate the performance of each optimisation approach in actual sample analysis under different Wasserstein ball radius settings. It is important to note that since the deterministic model, SP model, and ETO approach do not consider the Wasserstein ambiguity set, their performance remains unchanged with varying epsilon values and serves only as a benchmark for reference. These results demonstrate that the performance of the ETO and ETDRO approaches is markedly superior to that of the deterministic, SP, and DRO models. This suggests that by fully utilising historical and auxiliary data through ML methods to obtain more accurate estimates of the probability distribution of unknown aircraft arrival and departure times, we can frequently devise aircraft sequencing and scheduling plans that more closely align with actual scenarios. Furthermore, we also find that the performance of the SP and DRO models in actual sample analysis generally surpasses that of the deterministic model, suggesting that incorporating stochasticity in optimisation approaches can lead to relatively good decisions in actual scenarios.

In addition, we find that in practical scenarios, by properly selecting the $\epsilon$ value, the ETDRO approach can provide better performance than the ETO approach in aircraft

sequencing and scheduling. Specifically, when $\epsilon = 0$ for ETDRO, the two approaches are equivalent, with both having an objective value of 2,717.42. As the $\epsilon$ value of ETDRO increases from 100 to 1,000, its objective value strictly decreases from 2,693.48 to 2,604.04, resulting in a percentage reduction in the objective value compared to ETO increasing from 0.88% to 4.17%. However, as the $\epsilon$ value of ETDRO further increases from 1,000 to 5,000, its objective value subsequently increases from 2,604.04 to 2,648.11, and the percentage reduction in the objective value compared to ETO decreases from 4.17% to 2.55%. These results demonstrate that the ETDRO approach can enhance the performance of predetermined aircraft sequencing and scheduling plans in actual scenarios by considering potential prediction and sampling errors. It should be noted that the performance of the ETDRO approach in real-world scenarios initially improves with an increase in the $\epsilon$ value. Nevertheless, the performance of the ETDRO approach declines once the $\epsilon$ value surpasses a certain threshold. This indicates that the approach does not consistently improve with increasing $\epsilon$ values. Therefore, the $\epsilon$ value of the ETDRO approach must be carefully selected to ensure its best performance.

It is worth noting that the SP model outperforms the DRO model in practical scenarios when the probability distribution is estimated based solely on historical data of uncertain parameters. The objective value of the DRO model deteriorates as the $\epsilon$ value increases, performing worse than the deterministic model when the $\epsilon$ value is greater than or equal to 3,000. This may be because the probability distribution of uncertain parameters cannot be accurately estimated using only historical data. The DRO approach seeks to optimise the expected cost under the worst-case distribution among all possible distributions, exacerbating the inaccuracies in the already imprecise probability distribution and leading to suboptimal results in actual scenarios.

We present the detailed results of the actual sample analysis in Figure 4.6. The performance of makespan and average delay time aligns with the overall results, with both ETO and ETDRO approaches outperforming other optimisation approaches. Specif-

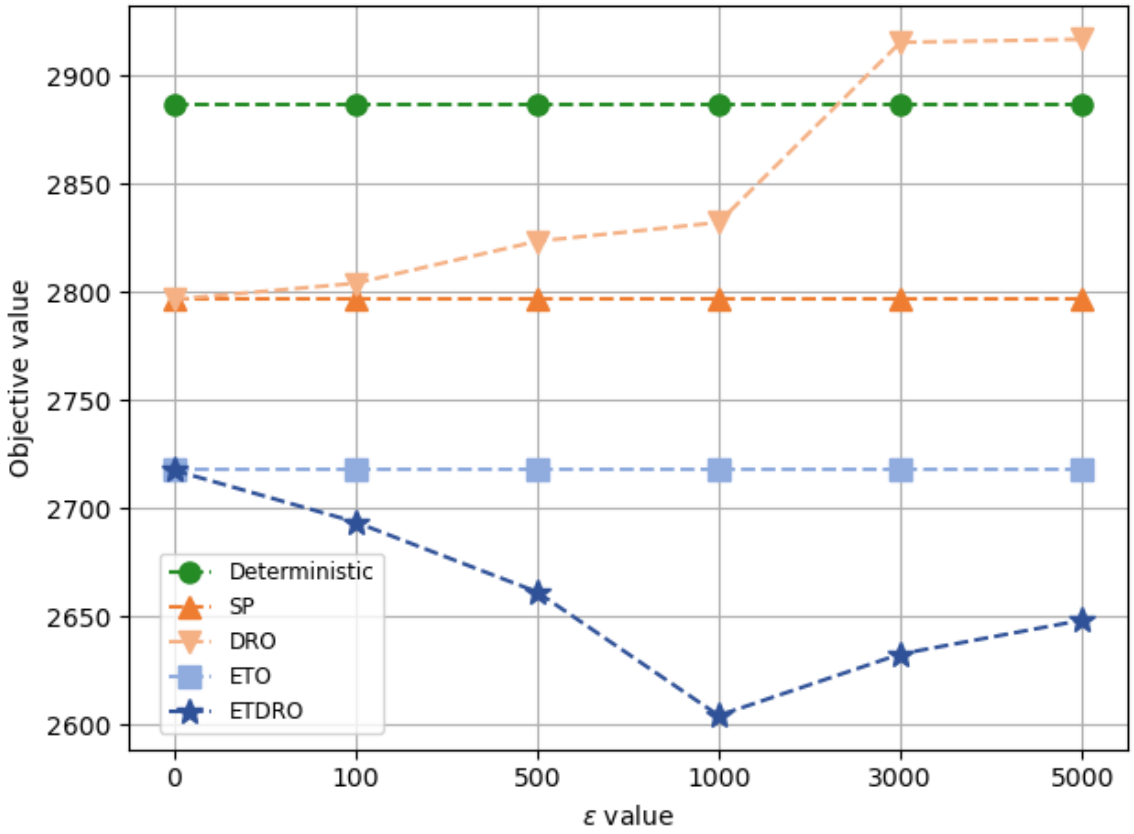


Figure 4.5: Overall results of actual sample analysis for ASSP.

ically, for the makespan metric, ETDRO achieves its best performance at an $\epsilon$ value of 3,000, recording a makespan of 1,280.10. This is roughly 1.25% lower than ETO' s 1,296.26, 1.38% below the SP model' s 1,298.02, 1.51% under the DRO model' s 1,299.73, and a significant 3.34% less than the deterministic model' s 1,324.39. For the average delay time metric, ETDRO performs optimally at an $\epsilon$ value of 1,000, achieving a delay time of 391.94. This constitutes a reduction of approximately 5.99% compared to ETO' s 416.93. Moreover, it shows a decrease of 7.76% relative to the SP model' s 424.91, an 8.16% decrease compared to the DRO model' s 426.79, and a substantial 12.33% reduction compared to the deterministic model' s 447.05. These results highlight the superior performance of the ETDRO approach in both makespan

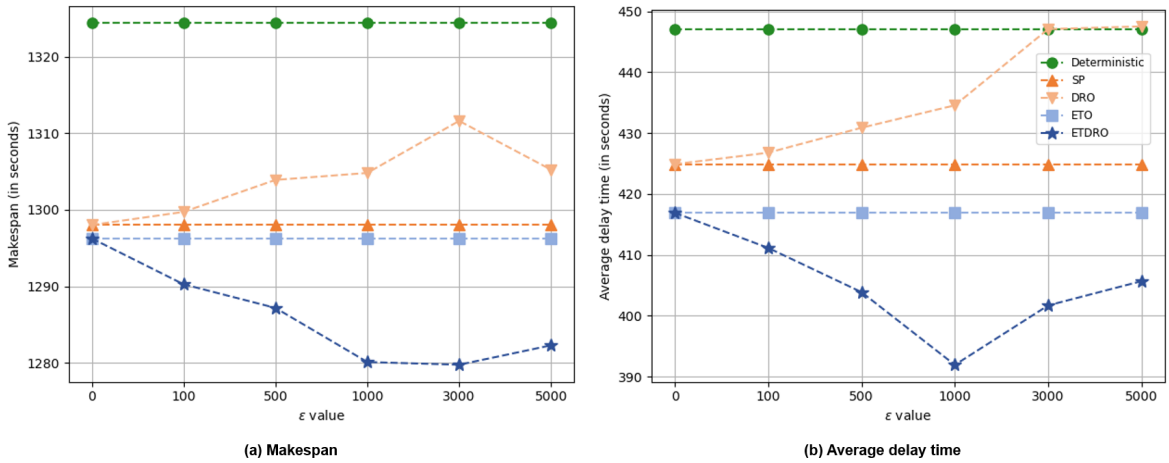


Figure 4.6: Detailed results of the actual sample analysis for ASSP.

and average delay time metrics under actual scenarios.

### 4.4.6 Real-world implementation for ASSP

In the real-world implementation, we consider the runway operations at XMN on 31st October 2023. The real-world implementation experiment employs a rolling horizon approach, with decision horizons set at 20-minute intervals, as outlined in the experimental design subsection. Specifically, to manage the expected arrivals and departures of aircraft, decision-making for ASSP is conducted 20 minutes in advance to generate sequencing plans for each decision horizon. For example, suppose the aircraft is scheduled to arrive or depart starting at 10:00. In that case, the decision-making occurs at 9:40. At this time, optimisation approaches are executed, considering the aircraft scheduled between 10:00 and 10:20. Upon completion, the aircraft sequencing plan for the 10:00 to 10:20 is fixed based on the obtained solution. At 10:00, with more precise information on arrival and departure times, the scheduling plan for aircraft between 10:00 and 10:20 is finalised. Afterwards, at 10:00, the subsequent optimisation run is performed, considering aircraft arriving and departing between 10:20 and 10:40. The plan for the previous period must be completed before executing the plan for the

Table 4.12: Results of the real-world implementation for ASSP.

| | Deterministic | SP | DRO | ETO | ETDRO |
|---|---|---|---|---|---|
| Total | 3,609.04 | 3,310.47 | 3,315.36 | 3,299.64 | **3,208.17** |
| Makespan | 1,404.41 | 1,346.39 | 1,348.11 | 1,346.32 | **1,332.39** |
| Average delay time | 576.37 | 526.83 | 527.86 | 525.71 | **508.74** |

subsequent time period.

We compare the deterministic, SP, and DRO models, along with the ETO and ETDRO approaches. The $\epsilon$ value for the DRO model is set to 100, while the $\epsilon$ value for the ETDRO approach is set to 1,000, as these parameter choices enable DRO and ETDRO to achieve their best performance in the actual sample analysis. For simplicity, we present the average result of all time periods in Table 4.12. The total objective value indicator of the ETDRO approach stands at 3,208.17, representing a reduction of 2.77% to 11.10% compared to other optimisation approaches. Specifically, ETDRO achieves a makespan of 1,332.39, which is 1.03% to 5.13% lower than the results from other approaches. Furthermore, the average delay time for ETDRO is 508.74, showing a decrease of 3.23% to 11.73% compared to alternatives. However, the total objective value, makespan, and average delay time of the ETO approach are 3,299.64, 1,346.32, and 525.71, respectively. These values are slightly lower than those of the deterministic, SP, and DRO models but are much higher than those of ETDRO. These results highlight that the ETDRO approach reduces makespan and delays by incorporating forecasting and sampling errors into decision-making. This is particularly important for runway controllers, who should prioritise the ETDRO approach in real-world applications to enhance the efficiency and accuracy of aircraft sequencing and scheduling plans.

### 4.4.7 The impact of the runway controller preference levels

We assign weights $W_1$ and $W_2$ to reflect the runway controller's preferences for makespan and delay time, respectively, where $W_1 + W_2 = 1$. The makespan and average delay time of the optimal decision under different preference levels are shown in Figure 4.7. As $W_1$ decreases and $W_2$ increases, indicating a decreased preference for makespan and an increased preference for delay time, the makespan rises from 5,138.96 to 5,153.99, an increase of 0.29%, while the average delay time decreases from 711.91 to 631.73, a reduction of 11.26%. The 0.29% increase in makespan indicates a slight compromise in overall operational efficiency while improving punctuality. The 11.26% reduction in average delay time demonstrates that prioritising delay time significantly enhances the on-time performance of individual aircraft. Although improving punctuality may slightly affect overall operational efficiency, this impact is acceptable in scenarios where reducing delays is of greater importance. Specifically, we also find that when runway controllers shift from focusing solely on makespan to incorporating a slight emphasis on delay time (with $W_2$ increasing from 0 to 0.05), the delay time can be significantly reduced by 10.44%. As the weight on delay time increases, the delay time reduction becomes more gradual.

In summary, runway controllers face a trade-off between overall operational efficiency (makespan) and individual aircraft punctuality (delay time). They must strike a delicate balance between the runway system's efficiency and individual aircraft's punctuality. By making informed weighing choices, runway controllers can effectively balance these two metrics while enhancing the efficiency and robustness of aircraft sequencing and scheduling plans.

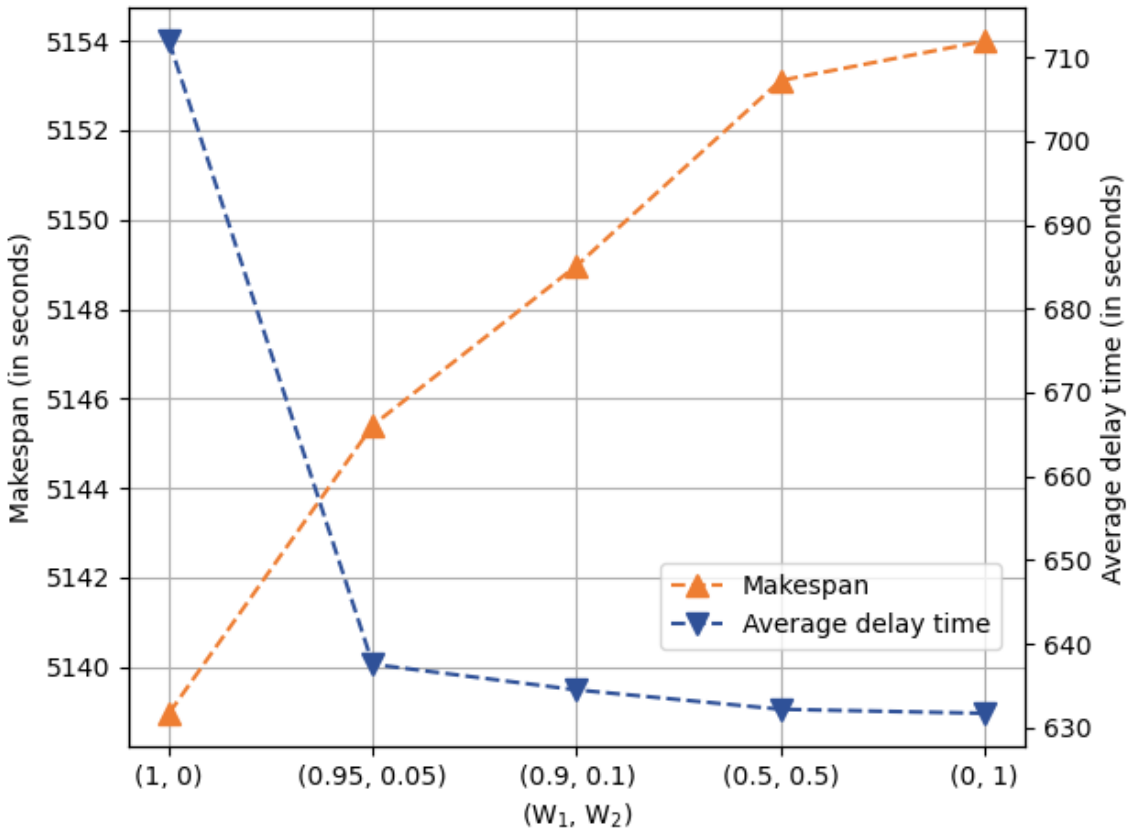


Figure 4.7: The impact of preference levels on makespan and average delay time.

### 4.4.8 Managerial implications and insights

The numerical experiments discussed above provide valuable managerial implications and insights for runway controllers. The following summary outlines the key findings derived from these experiments:

(i) Both in-sample and out-of-sample analyses reveal that the ETO approach and the ETDRO approach with smaller perturbations (i.e., smaller $\epsilon$ value) perform better in normal situations. Besides, the ETDRO approach consistently performs better than the ETO approach in extreme situations, regardless of the extent of the perturbation. Overall, the ETDRO approach with smaller perturbations consistently performs well

in normal and extreme scenarios, regardless of whether the data is known or unknown. This suggests that runway controllers should consider using the ETDRO approach with smaller $\epsilon$ values to balance performance in normal and extreme situations, thereby ensuring optimal operational efficiency and robustness.

(ii) According to the actual sample analysis, the ETO and ETDRO approaches are superior to other optimisation approaches. Additionally, the ETDRO approach outperforms the ETO approach by considering prediction and sampling errors in decision-making. However, it should be noted that the performance of the ETDRO approach in actual sample analysis does not continually improve with an increase in the radius of the Wasserstein ball. Instead, performance deteriorates beyond a certain threshold. Therefore, it is necessary to determine the appropriate $\epsilon$ value through numerical experiments. Moreover, the real-world implementation results demonstrate that the ETDRO approach significantly surpasses other optimisation approaches regarding makespan and delay time. The improvement of ETO compared with traditional optimisation approaches is not as pronounced as that of ETDRO.

(iii) The overall experimental results indicate that when historical data is available, the ETO and ETDRO approaches, enhanced by high-performance ML methods, significantly improve aircraft sequencing and scheduling decisions compared to traditional optimisation approaches. This implies that incorporating historical data and advanced ML techniques should be prioritised to enhance operational efficiency and decision-making processes. Moreover, since ETDRO offers a robust framework for dealing with uncertainty, particularly when the probability distribution is unknown or variable, its flexibility and robustness make it the preferred choice in many high-risk or highly uncertain environments. This approach demonstrates superior performance in in-sample, out-of-sample, and actual sample analyses, as well as real-world implementation compared to other optimisation approaches. Therefore, it is recommended that runway controllers prioritise the ETDRO approach in real-world runway operations to enhance

operational efficiency and reduce delays.

(iv) The analysis also underscores the importance of weight adjustment in balancing overall operational efficiency and individual aircraft punctuality. By carefully determining the weights for makespan and delay time, runway controllers can significantly reduce delays with a slight increase in makespan, while enhancing the robustness and efficiency of aircraft sequencing and scheduling.

## 4.5 Scalability analyses for ASSP

In this study, we propose exact and inexact decomposition methods for the ETDRO approach for ASSP. This section conducts the experiments on 56 test instances generated in Subsection 4.4.1. The runway controllers' preference levels for makespan and aircraft delay costs are set to $W_1 = 0.5$ and $W_2 = 0.5$. The performance indicators include the CPU time, objective value, and optimality gap. Solution methods are coded in Python with commercial MIP solver GUROBI. All the experiments are conducted on a computer with INTEL CORE i7-12700K 12 Core 20 Threads CPU @ 5.00 GHz and memory of 32 GB. We set the CPU time to 1,200 seconds for each test instance. For ease of display and comparison, we only provide the average performance of 56 test instances for each solution method at different epsilon values.

### 4.5.1 Performance evaluation of exact decomposition methods

We first compare the computational performance of the proposed E-DR L-shaped and E-DR-BBC methods with the DR L-shaped method used in previous literature. In terms of CPU time results provided in Figure 4.8, we find that the E-DR-BBC method performs the best, followed by the E-DR L-shaped method, while the DR L-shaped method performs the worst. Specifically, as the $\epsilon$ value of the ETDRO approach increases from 100 to 5,000, the CPU time required for the E-DR-BBC method increases

from 269.19 to 915.52 seconds, and for the E-DR L-shaped method from 492.53 to 1,018.52 seconds. The CPU time difference between the E-DR-BBC and E-DR L-shaped methods has decreased from over 200 seconds to 100 seconds. Additionally, the CPU time for the DR L-shaped method remains relatively stable at 1,000 seconds. The results demonstrate that the algorithmic enhancements in the E-DR L-shaped method can effectively reduce the CPU time required when the $\epsilon$ value is less than or equal to 1,000. However, when the $\epsilon$ value exceeds 1,000, these enhancements lose their effectiveness on the CPU time, and the performance becomes similar to that of the DR L-shaped method. Additionally, by adopting the BBC framework, the E-DR-BBC method further improves its CPU time performance and surpasses the other two exact decomposition methods regardless of the $\epsilon$ value.

Based on the overall results for the objective value indicator provided in Figure 4.9, we compare the objective value indicator and find that the E-DR-BBC and E-DR L-shaped methods perform similarly, both significantly outperforming the DR L-shaped method. Specifically, the performance of the E-DR-BBC and E-DR L-shaped methods is almost identical when the $\epsilon$ value of the ETDRO approach increases from 100 to 1,000. However, the E-DR-BBC method performs slightly better as the $\epsilon$ value increases further when $\epsilon$ is 3,000, the objective value of the E-DR-BBC method is reduced by 3.88% compared to the E-DR L-shaped method. As $\epsilon$ increases to 5,000, this reduction expands to 6.15%.

Figure 4.10 provides the overall results for the optimality gap indicator of the exact decomposition methods. The optimality gap of the E-DR L-shaped method increases from approximately 0.58% to 37.22% as the $\epsilon$ value increases from 100 to 5,000, while the optimality gap of the E-DR-BBC method increases from 1.16% to 31.47%. In comparison, the optimality gap of the DR L-shaped method remains stable at approximately 84%, indicating its difficulty in convergence. We find that the E-DR-BBC and E-DR L-shaped methods perform similarly and significantly outperform the

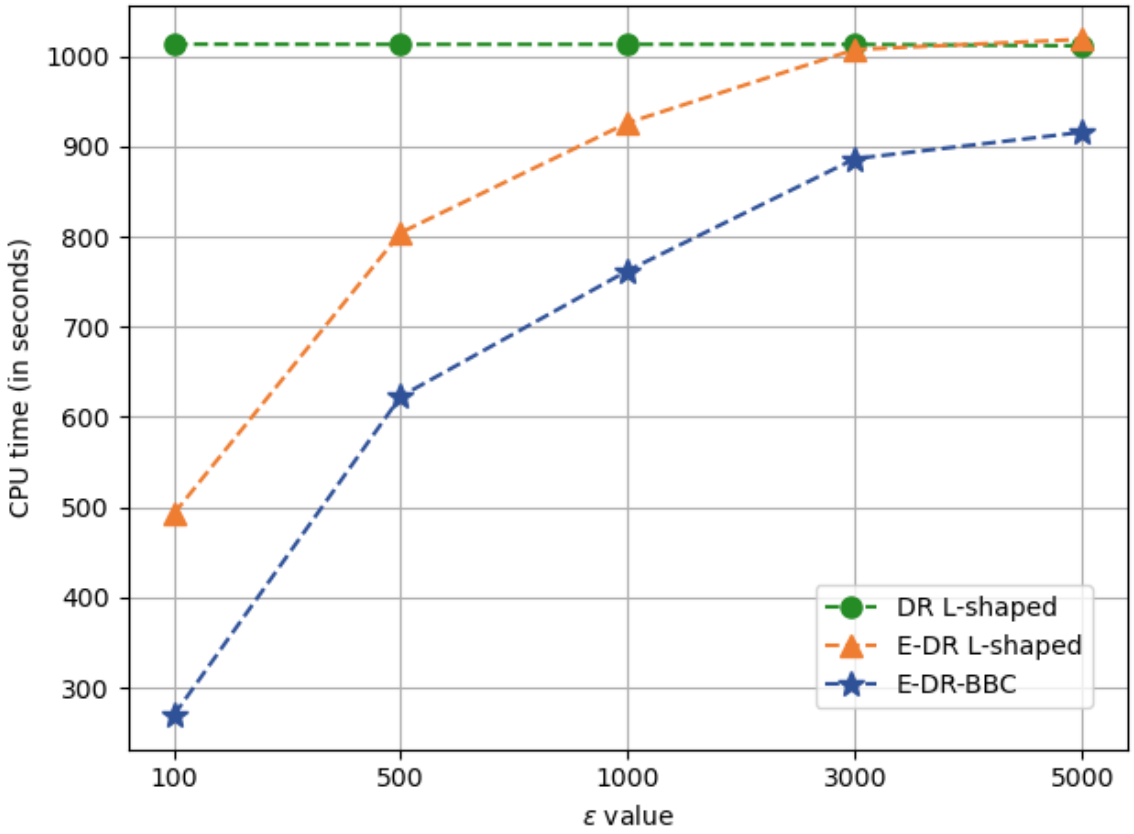


Figure 4.8: Overall results for the CPU time indicator of the exact decomposition methods for ETDRO.

DR L-shaped method. Specifically, when the $\epsilon$ value of the ETDRO approach increases from 100 to 3,000, the performance of the E-DR L-shaped method is slightly better than that of the E-DR-BBC method, with the reduced optimality gap fluctuating between 0.46% and 0.58%. However, as the value of $\epsilon$ increases further, the E-DR-BBC method demonstrates superior performance, with its advantage becoming more pronounced. When $\epsilon$ value is 3,000, the E-DR-BBC method achieves a reduction of 4.20 (14.82%) compared to the E-DR L-shaped method. As $\epsilon$ value increases to 5,000, this reduction grows to 5.75 (15.45%).

In summary, the comparison of exact decomposition methods demonstrates that

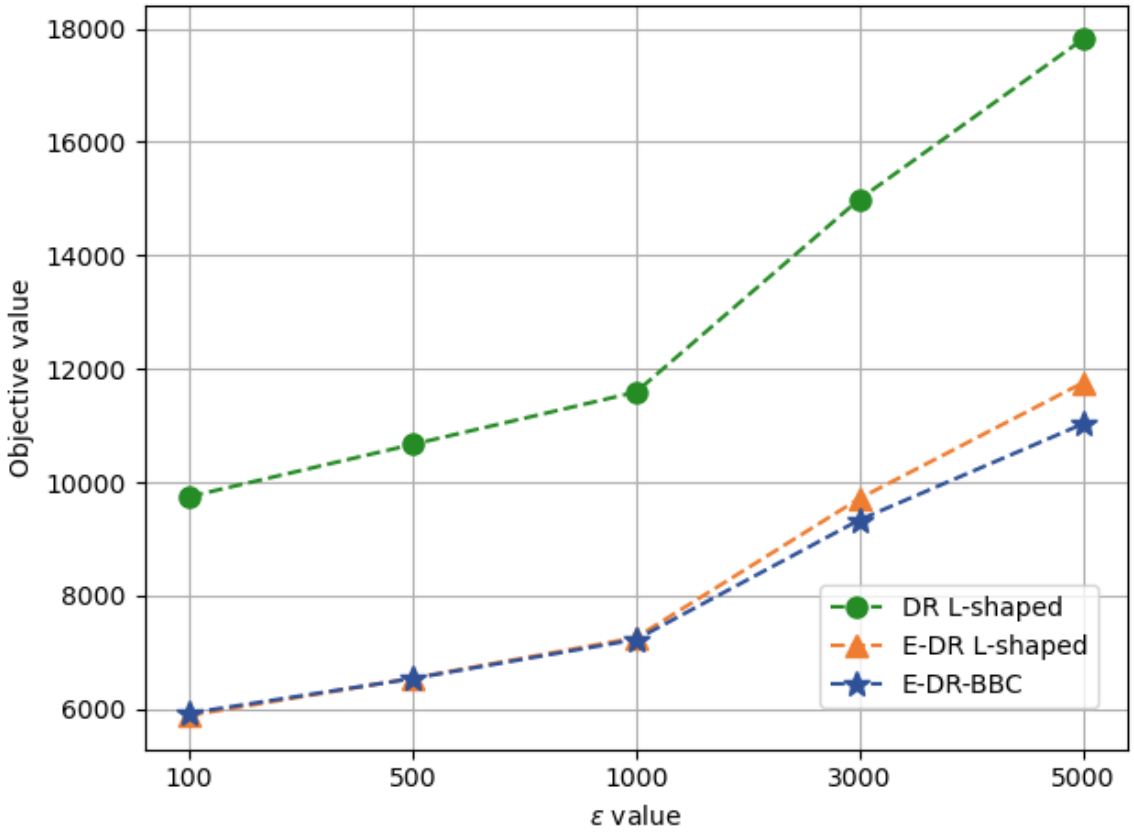


Figure 4.9: Overall results for the objective value indicator of the exact decomposition methods for ETDRO.

the E-DR-BBC method outperforms both the E-DR L-shaped and DR L-shaped methods across various metrics, including CPU time, objective value, and optimality gap. The E-DR-BBC method exhibits superior computational efficiency, with its advantage becoming more pronounced as the $\epsilon$ value increases. This indicates that the algorithmic enhancements, including the incorporation of time constraints, LBL cuts, and initial cuts, as well as the BBC framework, have played a practical and effective role in significantly improving computational performance.

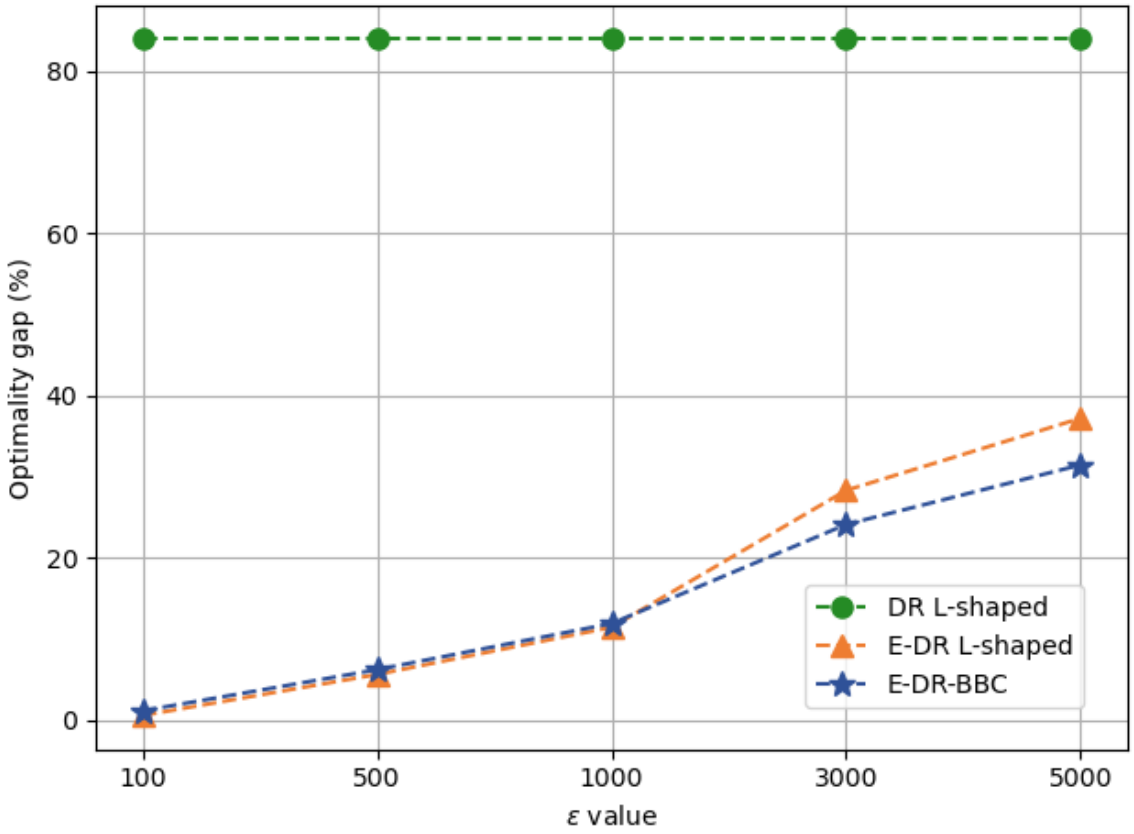


Figure 4.10: Overall results for the optimality gap indicator of the exact decomposition methods for ETDRO.

### 4.5.2 Performance evaluation of inexact decomposition methods

Among the exact decomposition methods, the E-DR-BBC method achieves the best performance. However, its CPU time increases from 269.19 to 915.52 as the $\epsilon$ value rises. Particularly, when applying the ETDRO approach with $\epsilon = 1000$, which performs best in actual sample analysis and real-world implementation, the CPU time required by the E-DR-BBC method is 762.25. For near real-time optimisation problems such as runway operations, despite providing slightly better solution quality, such a prolonged

CPU time is deemed unacceptable. Therefore, this paper further proposes the S-DR-BBC method to significantly reduce CPU time while ensuring that the solution quality remains within an acceptable range. In this subsection, we evaluate the impact of the trust region radius $k$ on the S-DR-BBC method. Previous studies have indicated that increasing the $k$ value beyond 3 becomes challenging to handle (Balakrishnan and Chandran, 2010; Prakash et al., 2021, 2022). Therefore, we primarily test $k$ values of 1, 2, 3, and $|I|$. It is important to note that when $k = |I|$, our S-DR-BBC method considers the complete solution space of the original problem, making the S-DR-BBC and E-DR-BBC methods equivalent.

We first compare the performance of inexact decomposition methods with different trust region radii $k$ in terms of CPU time as provided in Figure 4.11. Generally, as the value of $k$ increases, the required CPU time also increases accordingly. Specifically, we find that when $k = 1$, the CPU time slightly increases with the $\epsilon$ value in the ETDRO approach, from 49.11 to 71.23, yet all the calculations can still be completed within 100 seconds. For $k = 2, 3, |I|$, all their CPU time exceeds 190 seconds. As the $\epsilon$ value increases from 100 to 5,000, the CPU time for these $k$ values ranges from 195.09 to 283.92 seconds initially and increases to 535.30 to 906.09 seconds.

We then compare the performance of the inexact decomposition methods based on the objective value metric. The overall results for this metric are shown in Figure 4.12. We find that the loss, defined as the difference between the objective value of the solution obtained by the inexact method and that of the exact solution, grows as the $\epsilon$ value increases. Moreover, when the trust region radius $k$ is set to 1, which considers the smallest solution space, the highest loss value is obtained. As the value of $k$ increases, a larger solution space is considered, thereby reducing the loss. It is also observed that when the $\epsilon$ value of the ETDRO approach is between 100 and 1000, the performance of all solution methods is nearly identical. At an $\epsilon$ value of 3000, the loss between the inexact S-DR-BBC method with $k = 1$ and the exact E-DR-BBC

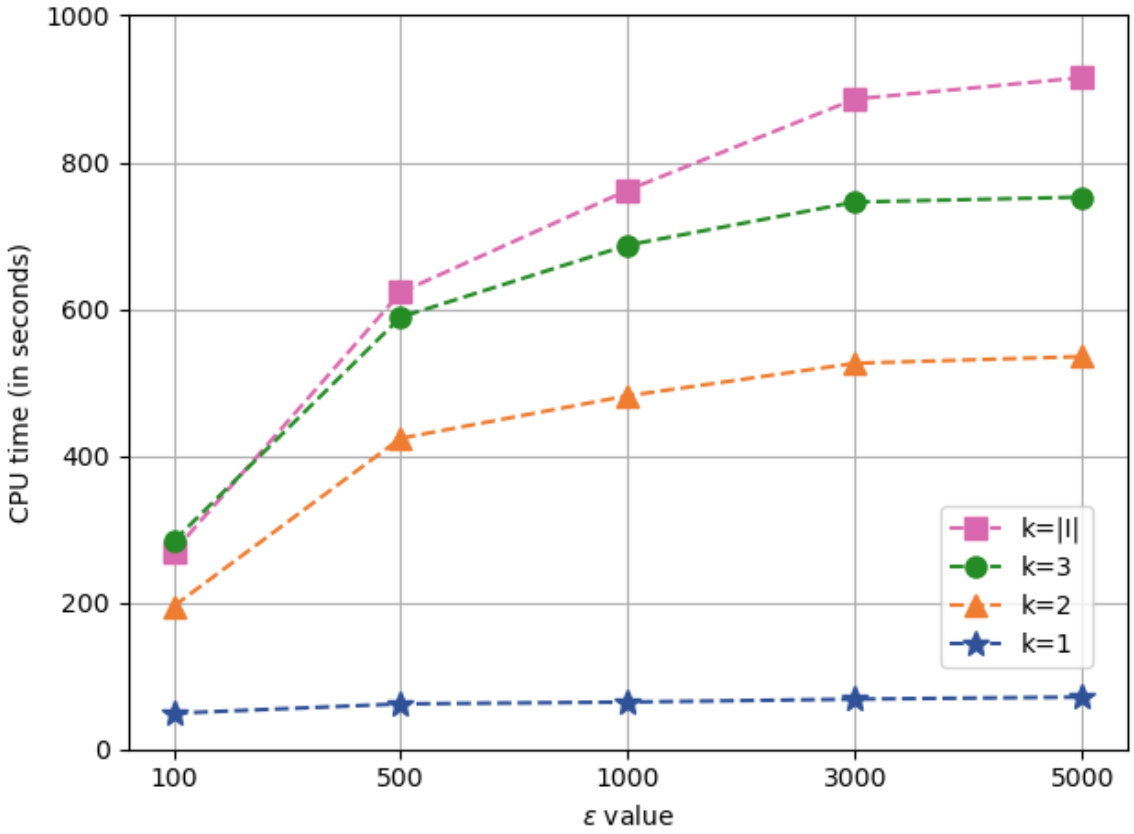


Figure 4.11: Overall results for the CPU time indicator of the inexact decomposition methods for ETDRO.

method is about 2.77%. When the $\epsilon$ value reaches 5000, the loss increases to about 4.39%. While the performance gap between the inexact S-DR-BBC method and the exact E-DR-BBC method increases with higher $\epsilon$ values, the results from actual sample analysis and real-world implementation indicate that increasing the $\epsilon$ value does not always enhance ETDRO performance in practical scenarios. Therefore, selecting an appropriate $\epsilon$ value for ETDRO, which is generally not large, is sufficient to keep the loss within an acceptable range. For instance, in the ETDRO approach for ASSP, the best performance in actual scenarios is achieved with an $\epsilon$ value of 1000. Notably, the loss between the inexact S-DR-BBC method with $k = 1$ and the exact E-DR-BBC

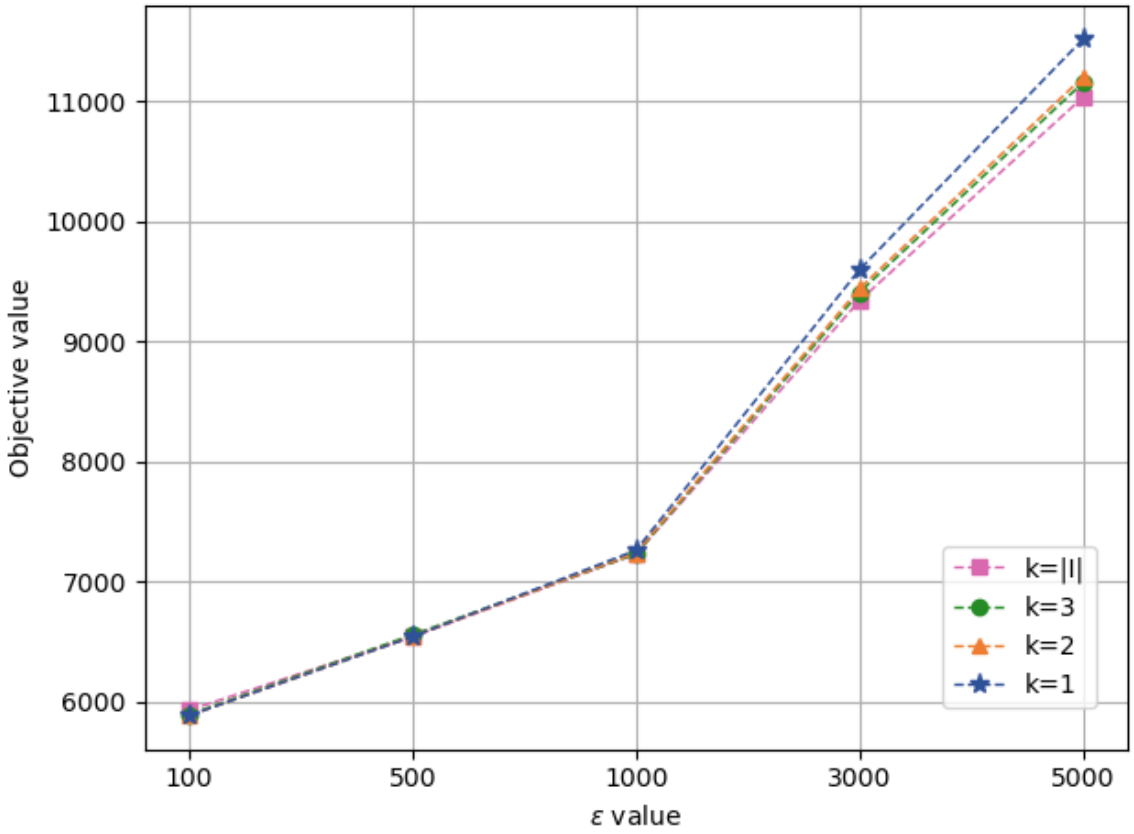


Figure 4.12: Overall results for the objective value indicator of the inexact decomposition methods for ETDRO.

method at this $\epsilon$ value is only approximately 0.43%, which is a minimal and acceptable loss, especially given the significant reduction in CPU time.

Lastly, we compare the performance of the inexact decomposition methods based on the optimality gap indicator. The overall results for this indicator are provided in Figure 4.13. It can be seen that for all test instances, the inexact S-DR-BBC method with $k = 1$ finds the optimal solution within its solution space. For other methods, the optimality gap is greater than 0, indicating a potential for further optimisation to narrow or eliminate this gap. Additionally, for $k = 2, 3, |I|$, as the $\epsilon$ value in the ETDRO approach increases, the optimality gaps also increase, ranging from 0.62% to

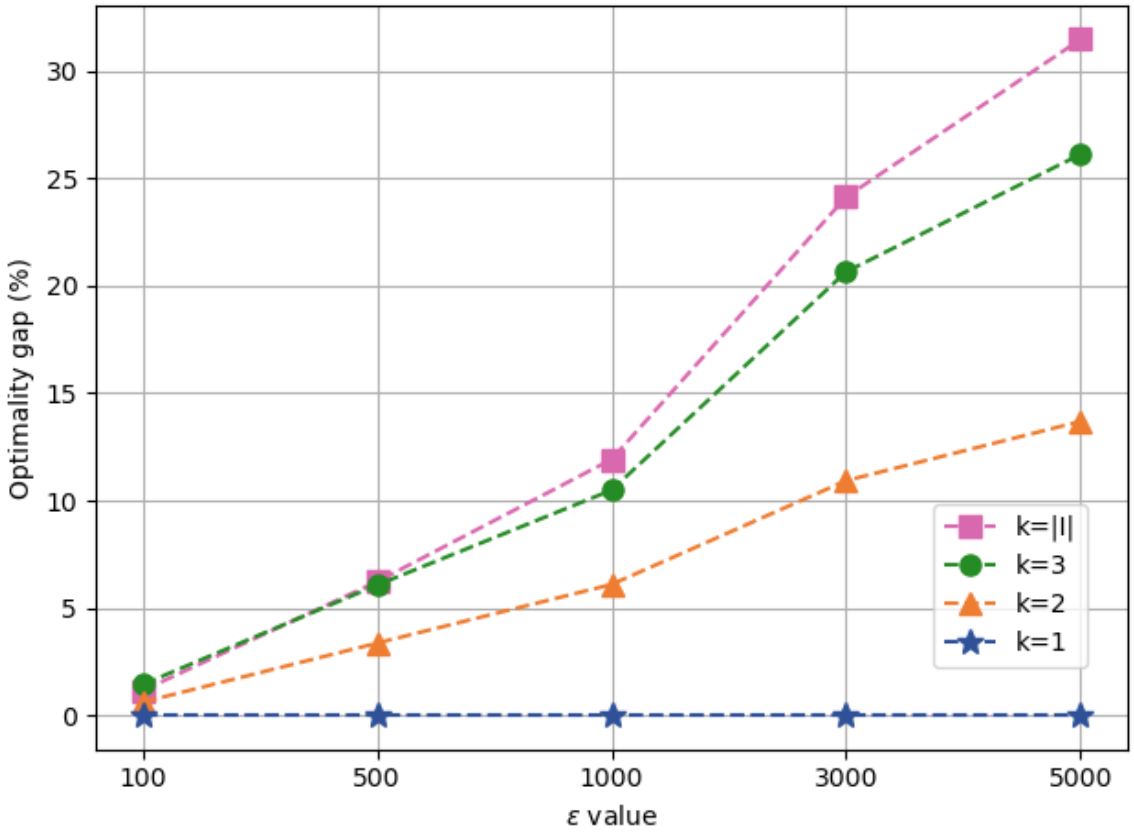


Figure 4.13: Overall results for the optimality gap indicator of the inexact decomposition methods for ETDRO.

1.44% initially and rising to 13.69% to 31.47%.

Overall, we find that the S-DR-BBC method with $k = 1$ outperforms other parameter settings in terms of CPU time and optimality gap indicators while maintaining acceptable loss in objective value. This suggests that setting $k = 1$ in the S-DR-BBC method is advantageous for implementing the ETDRO approach for ASSP in practical applications due to its high computational efficiency and excellent solution quality. This enables runway controllers to utilise the ETDRO approach to effectively manage the dynamic fluctuations of aircraft landing and take-off times across the entire planning horizon. This facilitates the rapid development of efficient and robust aircraft

sequencing and scheduling plans for real-world applications.

## 4.6 Conclusions

This study introduces advanced ETO and ETDRO approaches for ASSP at single-runway airports, aiming to enhance runway operations by integrating prediction and optimisation techniques for intelligent decision-making. We further develop several exact and inexact decomposition methods to handle the ETDRO approach efficiently. Extensive numerical experiments and scalability analyses using realistic data from XMN are used to evaluate the proposed optimisation approaches and solution methods. The in-sample and out-of-sample analyses reveal that both the ETO approach and the ETDRO approach with smaller $\epsilon$ values perform better under normal conditions, while the ETDRO approach excels in extreme situations. Notably, the ETDRO approach with a smaller $\epsilon$ value demonstrates consistent good performance in both normal and extreme scenarios. Meanwhile, actual sample analysis and real-world implementation demonstrate that the ETDRO approach outperforms other optimisation approaches when an appropriate $\epsilon$ value is chosen. Overall, experimental results confirm that when information beyond historical data of uncertain parameters is available, the ETO and ETDRO approaches, driven by high-performance ML methods, deliver superior aircraft sequencing and scheduling decisions compared to traditional optimisation approaches. Moreover, the ETDRO approach, which accounts for prediction and sampling errors, provides more efficient and robust plans than the ETO approach. The scalability analysis demonstrates that the proposed inexact S-DR-BBC method significantly outperforms exact solution methods in terms of CPU time, while maintaining good solution quality. This enables runway controllers to effectively utilise the ETDRO approach, capturing the dynamic fluctuations of aircraft arrival and departure times over the entire decision horizon, thereby facilitating the rapid development

of efficient and robust aircraft sequencing and scheduling plans.

In future research, we will continue to explore the application of prescriptive analytics in airports with multi-runway systems, as well as in jointly optimising runway operations and terminal manoeuvring area traffic flow management. By fully utilising historical and auxiliary data, with the support of ML technology, we can better capture the uncertainty in both airside operations and terminal airspace management. This helps avoid suboptimal plans, improve overall operational efficiency, reduce environmental impact, and better meet the growing demand for air traffic. However, given the more complex operating environment, ML methods may be impacted by numerous extreme values, potentially causing predicted scenarios to deviate from reality. To address this issue, we need to develop scenario selection strategies that can distinguish and eliminate these extreme scenarios. Moreover, for these more complex optimisation problems, it is necessary to develop tailored solution methods based on their specific operational characteristics, thereby more effectively providing runway and terminal airspace operation plans for practical applications.

# Chapter 5

# Prescriptive analytics for the multi-runway aircraft landing problem

## 5.1 Introduction

The rapid growth in aviation demand in recent years has further strained the already limited runway capacity at airports (Bennell et al., 2011; Lieder et al., 2015; Ikli et al., 2021; Messaoud, 2021; Chen et al., 2024). Optimising runway operations, instead of adhering to the first-come, first-served rule, can reduce the number of aircraft in holding patterns (Balakrishnan and Chandran, 2010; Lieder et al., 2015; Solak et al., 2018; Ikli et al., 2021). This not only increases runway capacity but also enhances the operational efficiency of the runway system. Additionally, making reasonable decisions about runway operations can bring significant economic and environmental benefits (Bennell et al., 2011; Sabar and Kendall, 2015; Chen et al., 2024). One critical aspect of achieving these improvements is addressing the RSP. The RSP is a crucial optimisation task in the aviation field, aiming to effectively manage airport runways to ensure the

safe and efficient operation of aircraft (Bennell et al., 2011; Ikli et al., 2021). In the RSP, wake vortex separation requirements between consecutive aircraft must be considered operational constraints, and each aircraft should land or take off within its designated time window. The objectives of the RSP are typically to minimise total delay costs and to minimise makespan (Lieder et al., 2015; Bennell, J A and Mesgarpour, Mohammad and Potts, Chris N, 2017; Hong et al., 2017). Additionally, given the substantial growth in air traffic has led to elevated emissions levels within the terminal airspace, which negatively impacts the environment and public health in the surrounding regions, fuel consumption costs and exhaust emission costs are also considered objectives for optimising runway operations (Sölveling et al., 2011; Bennell, J A and Mesgarpour, Mohammad and Potts, Chris N, 2017; Ikli et al., 2021; Chen et al., 2024). The ALP for arriving aircraft, the ATP for departing aircraft, and the ASSP that considers both arrivals and departures are the main branches of the RSP, which have been widely studied (Harikiopoulo and Neogi, 2010; Bennell et al., 2011; Lieder et al., 2015; Sabar and Kendall, 2015; Salehipour, 2020; Ikli et al., 2021; Messaoud, 2021).

The classical RSP, which uses deterministic aircraft arrival and departure times, has been extensively studied (Bennell et al., 2011; Solak et al., 2018; Ikli et al., 2021; Messaoud, 2021). However, factors such as severe weather, air traffic delay propagation, technical challenges, and security considerations introduce variability in aircraft arrival and departure times, resulting in uncertainty about the input data for the RSP (Solak et al., 2018; Ng et al., 2017, 2020; Ikli et al., 2021; Khassiba et al., 2022). Stochastic optimisation approaches have been employed to enhance the robustness of decision-making in runway operations. Solveling et al. (2011) utilised an SP approach to address the RSP with uncertain arrival and departure times. Heidt et al. (2016) applied various RO approaches to develop robust runway operation plans. Kapolke et al. (2016) incorporated the uncertainty of arrival and departure times into the RSP using RO and SP approaches. Ng et al. (2017) determined runway scheduling decisions

by evaluating the robustness of feasible solutions under worst-case scenarios, thereby ensuring robust decision-making. Solak et al. (2018) introduced SP models based on network and slot formulations for the RSP under uncertainty. Khassiba et al. (2020) and Khassiba et al. (2022) used the SP approaches to extend the ALP on a single runway, where initial approach fixes are considered.

These studies on RSP under uncertainty mainly focus on single runways or dual parallel runways that handle aircraft take-offs and landings separately. However, due to the substantial air traffic, many major international airports now feature multiple runways (Messaoud, 2021). Among these, airports with parallel runways often adopt segregated parallel operations, where runways are designated separately for arrivals and departures, thereby enhancing operational efficiency and safety. Examples include HKIA, Los Angeles International Airport (LAX), London Heathrow Airport (LHR), Shanghai Pudong International Airport (PVG), and Dubai International Airport (DXB), etc. In this mode, some runways are used exclusively for arrivals, while others are used solely for departures. Segregated parallel operations enhance operational efficiency, safety, and simplicity by eliminating the need for separate monitoring controllers, reducing aircraft interaction and missed approaches, simplifying the air traffic control environment, and lowering the risk of pilot errors due to incorrect instrument landing system selection. Compared to single runways or dual parallel runways, runway operations on parallel multi-runway systems not only determine the landing and take-off sequences and times but also assign aircraft to specific runways, which is more complex to solve (Kapolke et al., 2016; Hong et al., 2018; Lieder et al., 2015; Salehipour, 2020; Sabar and Kendall, 2015). To this end, this chapter investigates runway operations under uncertainty at airports with parallel multi-runway systems, with a particular focus on the MALP under arrival time uncertainty. Our approach, however, can also be readily extended to the MATP with uncertain departure times. ATC must make decisions regarding aircraft assignment, sequencing, and scheduling within

a parallel multi-runway system for aircraft landing operations. ATC typically determines aircraft-to-runway assignment before the aircraft enters the terminal airspace. Aircraft sequencing decisions are usually made when the aircraft is in the terminal airspace, while scheduling decisions are typically made when the aircraft enters the final approach phase. In the previous SP model for runway operations, decisions are based on a two-stage process (Solak et al., 2018; Khassiba et al., 2020). These studies often assume that aircraft arrival times are known when the aircraft enters the final approach phase. Consequently, aircraft sequencing decisions are usually addressed in the first stage, while aircraft scheduling decisions are handled in the second stage after the arrival times of aircraft are revealed. Since the second-stage problem of this SP model, which considers only the aircraft scheduling decision, is formulated as a linear programming (LP) problem with continuous variables, we refer to this SP model as the SP model with continuous recourse (SP-CR). With the support of advanced aviation technologies, including precise navigation systems, real-time data processing, accurate predictive analysis, and efficient communication systems, ATC can monitor the flight status of aircraft in real time and make relatively accurate arrival time predictions when aircraft are operating in terminal airspace. Consequently, ATC can typically make aircraft sequencing decisions based on these revealed arrival times. Therefore, we can incorporate aircraft sequencing decisions into the second stage of the SP model, thereby proposing a new SP model for MALP. This model assigns arriving aircraft to runways in the first stage. Subsequently, it makes sequencing and scheduling decisions for the aircraft assigned to each runway in the second stage. The decisions made using this method are usually more flexible and more suitable for complex environments that require high accuracy and real-time adjustments. Since the second-stage problem of this SP model, which considers aircraft sequencing and scheduling decisions, is formulated as a MILP problem, we refer to this SP model as the SP model with mixed-integer recourse (SP-MIR).

Previous studies on the RSP under uncertainty have primarily focused on deriving the distribution of aircraft arrival and departure times by analysing historical data (Solveling et al., 2011; Solak et al., 2018) or based on empirical knowledge (Khassiba et al., 2020, 2022). Although effective, these classical methods often rely on past data and experience and struggle to cope with the increasingly complex airside operation environment. In recent years, the rapid development of big data technology has provided new opportunities to address uncertainties in runway operations. When ATC has access to a substantial amount of historical and auxiliary data, it becomes possible to use ML methods to more accurately estimate the distribution of aircraft arrival and departure times (Bertsimas and Kallus, 2020; Tian et al., 2023c; Wang and Yan, 2023). ML methods can extract valuable insights from data, providing more accurate predictions than classical methods. Driven by these precise estimates, subsequent optimisation techniques can develop plans that better align with actual operations. This integration of prediction and optimisation, known as prescriptive analytics, facilitates informed decision-making based on the available data (Wang and Yan, 2023; Tian et al., 2023a,b,c).

The ETO approach is a promising method in prescriptive analytics (Yang et al., 2024). This approach first employs a learning-driven scenario generation (LSG) method to estimate the distribution and select appropriate scenarios. Subsequently, these sampled scenarios are inputs for the subsequent SP models. In this chapter, we employ the random forest (RF) method to estimate the distribution of aircraft arrival times. The RF method is a supervised learning technique (Breiman, 2001). Due to its high prediction accuracy, resistance to outliers, and good interpretability, the RF method has been widely used in prescriptive analysis research (Bertsimas and Kallus, 2020). Although the LSG method typically provides good decisions in real-world scenarios, it generates too many scenarios. Solving all the generated scenarios is difficult and usually requires a long CPU time. Since runway operations are typically near real-time optimisation

problems, solving the relevant optimisation problems is necessary quickly. Therefore, a suitable scenario selection strategy must be adopted to keep the model tractable while ensuring the quality of the solution. The LSG method typically employs the SAA to select an appropriate number of scenarios (Yang et al., 2024). Given that MALP is usually implemented at hub international airports, where the operational environment is usually complex, the aircraft arrival time predictions provided by ML methods may be affected by extreme values. However, previous scenario selection strategies do not account for the processing of these extreme scenarios. If they are directly used for scenario selection, some extreme scenarios may be input into the subsequent optimisation problem, resulting in unsatisfactory decisions. It is essential to implement measures to mitigate the occurrence of such situations. To this end, this chapter proposes an optimisation-enhanced LSG (OLSG) method. This method selects a subset of scenarios from an initial scenario set using a p-median problem, with the objective function minimising the Wasserstein distance between the chosen scenarios (Reese, 2006; Wang and Jacquillat, 2020). By minimising the Wasserstein distance, extreme scenarios can largely be avoided in decision-making, ensuring that the decisions are more aligned with reality.

The ALP is recognised as an NP-hard problem (Bennell et al., 2011). When considering MALP under uncertainty, which involves multiple runways and uncertain aircraft arrival times, solving it to optimality becomes even more challenging. The studied MALP includes three categories of decision variables: aircraft-to-runway assignments, aircraft sequencing, and scheduling. This structure is well-suited to decomposition. The BD method is extensively used to solve SP models where the subproblems are required to be LP models with continuous variables (Khassiba et al., 2020; Rahmaniani et al., 2018). However, the studied SP-MIR model for MALP considers the aircraft sequencing and scheduling problem in the second stage, which is a MILP second-stage problem. For this case, the integer L-shaped method can be employed instead of the

classic BD method (Laporte and Louveaux, 1993). Nevertheless, in the integer L-shaped method, a continuous relaxation of the MILP subproblem is used to generate classical Benders cuts, which may be time-consuming (Elçi and Hooker, 2022). As a more efficient alternative method, the logic-based Benders decomposition (LBBD) method was widely applied in recent studies, where no-good cuts or analytical cuts are used as Benders cuts (Elçi and Hooker, 2022; Guo and Zhu, 2023; Li et al., 2023). Elçi and Hooker (2022) observed that the LBBD method generally delivers more favourable computational performance compared to the integer L-shaped method. Moreover, the branch-and-check (BAC) method, a variant of the LBBD method, can achieve even better computational performance (Elçi and Hooker, 2022; Li et al., 2023). The LBBD method only generates Benders cuts through the optimal solution of the master problem (Hooker, 2007). Different from the LBBD method, in the BAC method proposed by Thorsteinsson (2001), when an integer feasible solution of the master problem is identified, the master problem terminates, and the feasible solution is input into the subproblems to generate Benders cuts. An improved version is then proposed, wherein the BAC method operates within a branch-and-cut tree. Once an integer feasible solution is found, Benders cuts are generated and added to the nodes that remain to be explored (Beck, 2010; Tran et al., 2016). As the BAC method generates Benders cuts for every integer feasible solution found in the branch-and-cut tree, the number of Benders cuts produced during the search process can be substantial (Fachini and Armentano, 2020). Some weak Benders cuts may be added during this process, potentially slowing the convergence rate. To address this issue, we propose the stabilised BAC (SBAC) method, which introduces the trust region constraints and the reverse local branching constraints to the master problem. The motivation for the SBAC method is to stabilise the master problem around the neighbourhood of a stability centre point (i.e., a good feasible solution) to generate strong Benders cuts during the convergence process (Baena et al., 2020; Gong and Zhang, 2022).

The main contributions of this chapter are summarised as follows:

(i) We develop the SP-MIR model for MALP, aiming to devise efficient and environmentally friendly aircraft landing operations for a multi-runway system under uncertainty. This model minimises the weighted sum of expected makespan, fuel consumption and exhaust emission costs. For the SP-MIR model of the MALP, we employ the LSG method to estimate the distribution of unknown parameters using ML methods and generate scenarios based on the estimated results. Additionally, we propose an OLSG method for generating scenarios when the prediction results of the ML method perform poorly due to system complexity. This is achieved by using the p-median problem with the objective function of minimising the Wasserstein distance of the selected scenarios, thereby avoiding generating extreme scenarios as much as possible.

(ii) We propose a novel exact decomposition method called the SBAC method for efficiently solving the SP-MIR model for MALP. This method decomposes the original problem into a stabilised master problem for determining aircraft-to-runway assignments, as well as several subproblems for making aircraft sequencing and scheduling decisions. The stabilised master problem searches for new solutions around the neighbourhood of a stability centre point through trust region constraints and reverse local branching constraints, thereby generating strong Benders cuts to accelerate the convergence rate.

(iii) Extensive numerical experiments based on real data from Hong Kong International Airport (HKIA) demonstrate the efficiency and environmental benefits of the SP-MIR model for MALP, supported by the OLSG method. Additionally, scalability analyses evaluate the computational performance of the proposed SBAC method, showing a significant improvement in CPU time.

The rest of this chapter is structured as follows. Section 5.2 provides the problem setting and presents the formulations of the SP-MIR model for MALP. Section 5.3 illustrates the scenario generation methods used in this study. Section 5.4 introduces

the proposed SBAC method to solve the SP-MIR model for MALP. In Section 5.5, numerical experiments are performed based on realistic data from HKIA. The results of the scalability analyses are reported in Section 5.6. Finally, the conclusions are presented in Section 5.7.

## 5.2 SP-MIR model for MALP

In this section, we provide a problem description of the MALP under arrival time uncertainty and use the SP approach to handle this uncertainty. At airports with multi-runway systems for aircraft landing operations, ATC must make decisions regarding aircraft assignment, sequencing, and scheduling. Air traffic control is responsible for assigning landing runways and planning routes to each arriving aircraft before they enter terminal airspace. Once an aircraft is flying in terminal airspace, air traffic control can monitor its flight status in real time, make relatively accurate arrival time predictions, and determine aircraft sequencing and scheduling decisions for each runway based on the revealed arrival times. Therefore, the SP model for MALP involves a two-stage decision-making process. In the first stage, we assign aircraft to runways to optimise the utilisation of runway resources. In the second stage, we make sequencing and scheduling decisions for the arriving aircraft assigned to each runway to ensure they can land safely within the appropriate time windows. Since the second-stage problem of this SP model, which considers aircraft sequencing and scheduling decisions, is formulated as an MILP problem, we refer to this SP model as the SP-MIR model. Consider a set $R$ containing parallel, identical, independent runways for landing. The approaching aircraft are in a set $I$. The variable $x_i^r$ is equal to 1 if aircraft $i \in I$ is assigned to runway $r \in R$. Due to various factors, the arrival time of each aircraft is often uncertain. This uncertainty is represented by a potentially small finite set of scenarios, denoted as $\Omega = \{1, 2, \ldots, |\Omega|\}$. The probability of scenario $\omega \in \Omega$ is denoted

by $p^{\omega}$. The arrival time $E_i^{\omega}$ of each aircraft $i \in I$ is determined under scenario $\omega \in \Omega$. The landing time $t_i^{\omega}$ of aircraft $i$ must not be earlier than its arrival time $E_i^{\omega}$, and the delay time $d_i^{\omega}$ is the difference between these two values. Each runway can process, at most, one aircraft at a time. The variable $y_{ij}^{r\omega}$ is equal to 1 if aircraft $i \in I$ precedes aircraft $j \in I \setminus \{i\}$ on the same runway $r$ under scenario $\omega$. If aircraft $i$ and $j$ are assigned to the same runway, a separation time $S_{ij}$ is required between the landing times of aircraft to ensure the wake vortex separation requirements. Each runway starts with dummy aircraft $s$ and ends with dummy aircraft $e$. The arrival time of the earliest arriving aircraft under scenario $\omega$ is denoted as $A^{\omega}$, where $A^{\omega} = \min_{i \in I} \{E_i^{\omega}\}$. The variable $z^{\omega}$ represents the makespan under scenario $\omega$, defined as the difference between the landing time of the last aircraft to use the runway system and $A^{\omega}$.

The SP-MIR model for MALP aims to minimise the weighted sum of the expected makespan and environmental costs. In this chapter, weights $W_1$ and $W_2$ are respectively used to reflect the preferences for makespan and environmental costs, where $W_1 + W_2 = 1$. The environmental costs include fuel consumption and exhaust emission costs. The fuel consumption cost depends on the jet fuel cost per unit $C^{\text{Fuel}}$, the fuel burn rate $\alpha_i$, and the delay time $d_i^{\omega}$ (Sölveling et al., 2011). The extra fuel consumption cost of aircraft $i$ is expressed as $C^{\text{Fuel}} \alpha_i d_i^{\omega}$ for each scenario $\omega$. The exhaust emissions mainly consist of $CO_2$, CO, HC, $NO_x$, and $SO_2$ (Sölveling et al., 2011; Tian et al., 2018). Following the exhaust emission modelling method proposed by Sölveling et al. (2011), the emissions of $CO_2$ are proportional to the fuel flow $\alpha_i d_i^{\omega}$ with a factor $\beta$. The emissions of the other pollutants can be calculated by multiplying the emission rates $\varepsilon_i^m$, $m \in \{\text{CO}, \text{HC}, \text{NO}_x, \text{SO}_2\}$ by the delay time $d_i^{\omega}$. Each type of emission has an associated external cost, denoted as $C^{\text{CO}_2}$, $C^{\text{CO}}$, $C^{\text{HC}}$, $C^{\text{NO}_x}$, and $C^{\text{SO}_2}$. We provide notations of sets, parameters and decision variables used in Table 5.1.

Given the problem description and notations outlined above, the SP-MIR model

Table 5.1: Notations and definitions for MALP.

| Notation | Definition |
|---|---|
| **Sets** | |
| $R$ | Set of runways. |
| $I$ | Set of arriving aircraft. |
| $\Omega$ | Set of scenarios. |
| **Parameters** | |
| $p^\omega$ | Probability of scenario $\omega$. |
| $E_i^\omega$ | Arrival time of aircraft $i$ under scenario $\omega$. |
| $A^\omega$ | The arrival time of the earliest aircraft under scenario $\omega$. |
| $S_{ij}$ | The separation time between aircraft $i \in I$ and $j \in I \setminus \{i\}$. |
| $\alpha_i$ | The fuel burn rate of aircraft $i$. |
| $\beta$ | The proportional constant for $CO_2$ emissions per unit of fuel flow. |
| $C^{\text{Fuel}}$ | Unit cost of jet fuel. |
| $C^m$ | Unit cost of $m$, where $m \in \{CO_2, CO, HC, NO_x, SO_2\}$. |
| $\varepsilon_i^m$ | Emission rate of $m$ for aircraft $i$, where $m \in \{CO, HC, NO_x, SO_2\}$. |
| $W_1$ | The weight of the makespan. |
| $W_2$ | The weight of the environmental costs. |
| $M^\omega$ | A sufficiently large number for the scenario $\omega$. |
| **Variables** | |
| $x_i^r$ | 1, if aircraft $i$ is assigned to runway $r \in R$; 0, otherwise. |
| $y_{ij}^{r\omega}$ | 1, if aircraft $i$ precedes aircraft $j$ on the same runway $r$ under scenario $\omega$; 0, otherwise. |
| $t_i^\omega$ | Landing time of aircraft $i$ in scenario $\omega$. |
| $z^\omega$ | Makespan under scenario $\omega$. |
| $d_i^\omega$ | Delay time of aircraft $i$ in scenario $\omega$. |

for MALP is provided as follows:

$$\min \sum_{\omega\in\Omega} p^\omega \left[ W_1 z^\omega + W_2 \left( C^{\text{Fuel}} \sum_{i\in I} \alpha_i d_i^\omega + C^{\text{CO}_2} \sum_{i\in I} \beta\alpha_i d_i^\omega + C^{\text{CO}} \sum_{i\in I} \varepsilon_i^{\text{CO}} d_i^\omega \right.\right.$$
$$\left.\left. + C^{\text{HC}} \sum_{i\in I} \varepsilon_i^{\text{HC}} d_i^\omega + C^{\text{NO}_x} \sum_{i\in I} \varepsilon_i^{\text{NO}_x} d_i^\omega + C^{\text{SO}_2} \sum_{i\in I} \varepsilon_i^{\text{SO}_2} d_i^\omega \right) \right] \tag{5.1a}$$

$$\text{s.t.} \quad \sum_{r \in R} x_i^r = 1, \quad \forall i \in I, \tag{5.1b}$$

$$x_i^r = \sum_{j \in I \cup \{e\} \setminus \{i\}} y_{ij}^{r\omega}, \quad \forall r \in R, \forall i \in I, \forall \omega \in \Omega, \tag{5.1c}$$

$$\sum_{j \in I} y_{sj}^{r\omega} = 1, \quad \forall r \in R, \forall \omega \in \Omega, \tag{5.1d}$$

$$\sum_{i \in I} y_{ie}^{r\omega} = 1, \quad \forall r \in R, \forall \omega \in \Omega, \tag{5.1e}$$

$$\sum_{j \in I \cup \{s\} \setminus \{i\}} y_{ji}^{r\omega} = \sum_{j \in I \cup \{e\} \setminus \{i\}} y_{ij}^{r\omega}, \quad \forall r \in R, \forall i \in I, \forall \omega \in \Omega, \tag{5.1f}$$

$$t_i^\omega \geq E_i^\omega, \quad \forall i \in I, \forall \omega \in \Omega, \tag{5.1g}$$

$$t_i^\omega + S_{ij} - t_j^\omega \leq M^\omega (1 - y_{ij}^{r\omega}), \quad \forall r \in R, \forall i \in I, \forall j \in I, i \neq j, \forall \omega \in \Omega, \tag{5.1h}$$

$$z^\omega \geq t_i^\omega - A^\omega, \quad \forall i \in I, \forall \omega \in \Omega, \tag{5.1i}$$

$$d_i^\omega \geq t_i^\omega - E_i^\omega, \quad \forall i \in I, \forall \omega \in \Omega, \tag{5.1j}$$

$$x_i^r \in \{0, 1\}, \quad \forall r \in R, \forall i \in I, \tag{5.1k}$$

$$y_{ij}^{r\omega} \in \{0, 1\}, \quad \forall r \in R, \forall i \in I \cup \{s\}, \forall j \in I \cup \{e\}, i \neq j, \forall \omega \in \Omega, \tag{5.1l}$$

$$t_i^\omega \in \mathbb{R}^+, \quad \forall i \in I, \forall \omega \in \Omega, \tag{5.1m}$$

$$z^\omega \in \mathbb{R}^+, \quad \forall \omega \in \Omega, \tag{5.1n}$$

$$d_i^\omega \in \mathbb{R}^+, \quad \forall i \in I, \forall \omega \in \Omega. \tag{5.1o}$$

The objective function (5.1a) minimises the weighted sum of the expected makespan and expected environmental costs. Constraints (5.1b) ensure that each aircraft is assigned to exactly one runway. Constraints (5.1c) ensure each aircraft is within the runway sequence. Constraints (5.1d) and (5.1e) guarantee that each runway starts with dummy aircraft $s$ and ends with dummy aircraft $e$. Constraints (5.1f) maintain the flow conservation of each runway. Constraints (5.1g) require that the landing time of aircraft $i$ should be no earlier than $E_i^\omega$. Constraints (5.1h) ensure the separation time between two aircraft that use the same runway consecutively, where $M^\omega = \max_{i\in I}\{E_i^\omega\} + \max_{i\in I, j\in I,\, i\neq j}\{S_{ij}\}(|I|-1)$. Constraints (5.1i) compute the makespan. Constraints (5.1j) define the delay time of each aircraft. Constraints (5.1k) to (5.1o) define the domain of decision variables.

## 5.3 Scenario generation methods for MALP

In the SP approach, appropriate scenario generation methods are crucial. They generate scenarios that better capture uncertainty, enhancing the robustness and reliability of the SP models' decisions, thereby making them more effective in practical implementations. In this section, we first introduce the HDSG method, a scenario generation method based on historical data, which has been used in previous SP approaches for runway operations (Sölveling et al., 2011; Solak et al., 2018). Next, we propose the LSG method based on the RF method. In this approach, the RF method determines the distribution of aircraft arrival times, and an appropriate number of scenarios are extracted using the SAA method. Finally, based on the LSG method, we further develop the OLSG method. Given that the poor predictive accuracy of some decision trees may lead to significant deviations between the sampled and actual scenarios, the OLSG method employs the p-median problem to select scenarios generated by the RF method. This method prevents extreme scenarios from being incorporated into the

subsequent SP models.

### 5.3.1 Historical data-driven scenario generation method

In previous studies on runway operations under uncertainty using the SP approach, the HDSG method has often been employed to generate scenarios based on historical data (Sölveling et al., 2011; Solak et al., 2018). In this study, the HDSG method first analyses the deviation between the estimated and actual aircraft arrival times using historical data. Subsequently, it employs the SAA method to generate a scenario set $\Omega$ based on the analysed distribution, which is then utilised in the subsequent SP-MIR model for MALP.

Utilising data from 1st to 30th October 2023, we present the distribution of deviations between estimated and actual arrival times in Figure 5.1. Specifically, Figure 5.1(a) shows the distribution for all deviations, where the deviation times are concentrated around 0 minutes. However, numerous extreme values result in the distribution having a long tail. We eliminate these extreme values using the interquartile range (IQR) method based on the 0.25 and 0.75 quantiles. After removing outliers, Figure 5.1(b) shows a more symmetric distribution of arrival time deviations. In the HDSG method, a scenario is constructed by incorporating the arrival time deviations $\hat{a}_i$ into the known estimated arrival time $E_i$ for each aircraft $i$. These deviations, $\hat{a}_i$, are randomly drawn from the distribution shown in Figure 5.1(b). All sampled scenarios are included in the scenario set $\Omega$. The probability of each scenario in set $\Omega$ is $\frac{1}{|\Omega|}$.

### 5.3.2 Learning-driven scenario generation method

This section presents the LSG method, with its framework illustrated in Figure 5.2. To train the RF method, we start by presenting the experimental data and selected features. Then, we adjust the hyperparameters of the RF method and check how

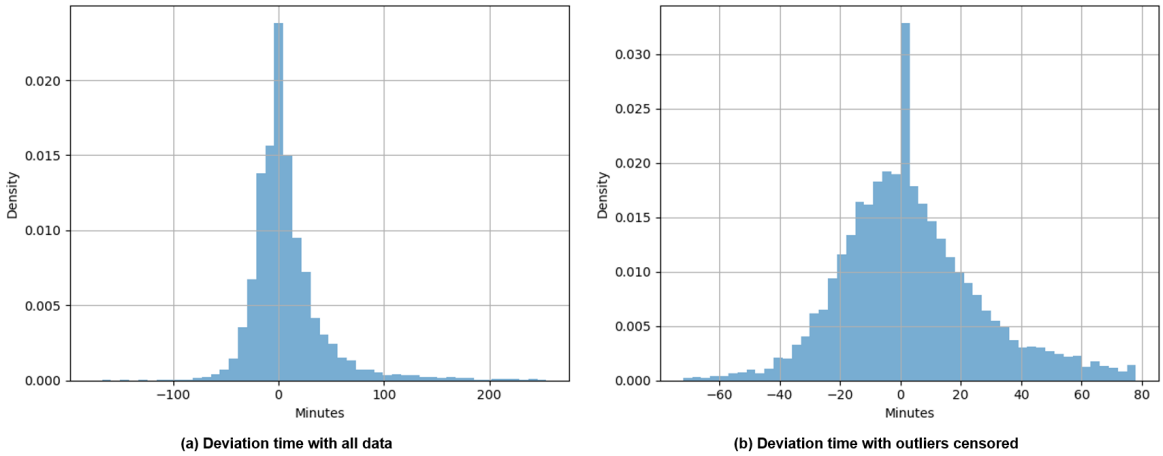


Figure 5.1: Distribution of arrival time deviations for MALP.

well it predicts aircraft arrival times. The details of the training process are provided below. Directly predicting aircraft arrival times can be affected by various uncertain factors. By predicting the deviation between estimated and actual arrival times instead, we can more effectively account for these uncertainties and enhance the accuracy of our predictions. Specifically, we predict the deviation value and add this predicted deviation, $\hat{a}_i$, to the estimated arrival time, $E_i$, of aircraft $i$ to provide the predicted arrival time $\hat{E}_i$, i.e., $\hat{E}_i = E_i + \hat{a}_i$.

The RF method applied in our study is implemented with the scikit-learn library. The dataset used in this study was collected from HKIA between 1st and 31st October 2023. It comprises 12,822 records of arriving aircraft, each containing relevant aircraft information. The TAF data includes details such as maximum and minimum temperatures, humidity, air pressure, wind direction, wind speed, and other relevant parameters during the forecast period. We selected 14 features from the original dataset, which are presented in Table 5.2, along with their data type, encoding method, and statistical information. Missing values were identified in features such as "Aircraft type", "Route distance", and "Fuel load". To address these, we filled in the missing values using the median for "Aircraft type" and the mean for both "Route distance" and "Fuel

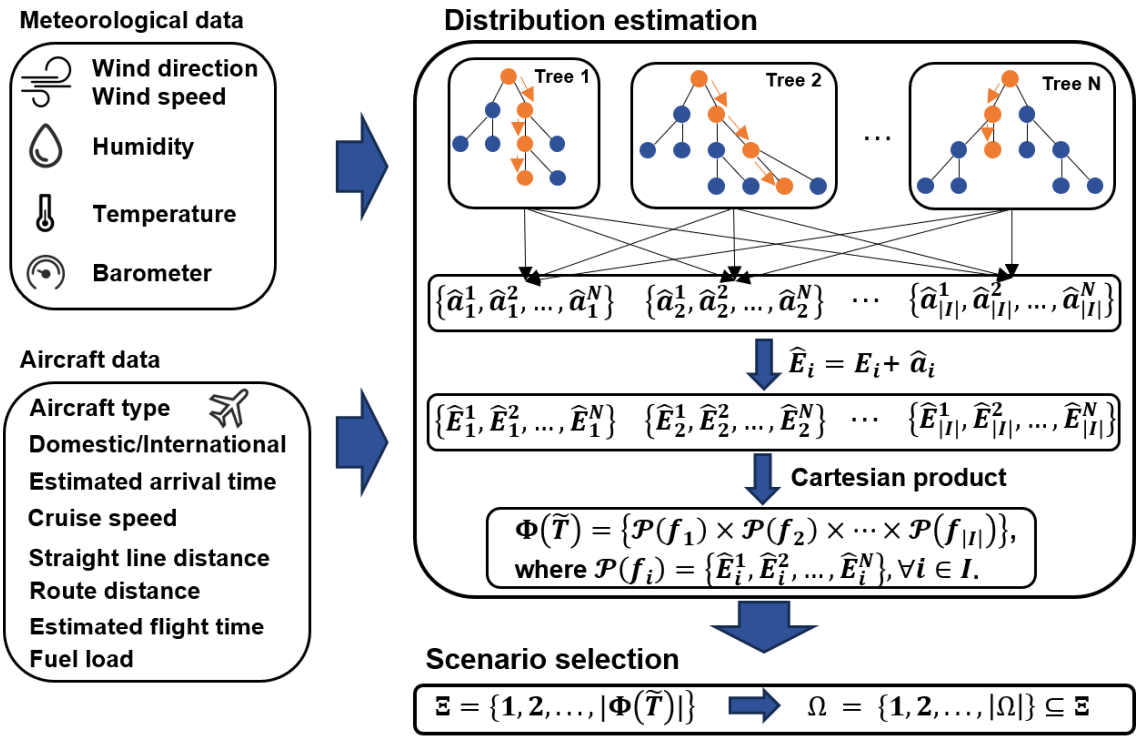


Figure 5.2: Framework for the scenario generation method supported by RF.

load". Notably, the features "Aircraft type", "Domestic/International", and "Wind direction" are expressed in a literal format, necessitating their conversion to numerical data. Regarding aircraft type, the arriving aircraft at HKIA primarily fall into two categories: heavy and large, which we encoded as 2 and 1, respectively. We considered 16 wind directions and adopted one-hot encoding for wind directions. The data collected between 1st and 30th October 2023 was designated as dataset $D$, containing 12,362 records of arriving aircraft. We randomly split 80% of the data in $D$ as a training set $D^{\text{Training}}$ for training the RF model, while the remaining 20% as $D^{\text{Testing}}$ to evaluate its performance. The data from October 31st 2023, is used to generate test instances for the numerical experiments and scalability analysis.

In the RF method, several key hyperparameters must be considered. The hyperparameters tuned for the RF method are detailed in Table 5.3. These optimal hyperpa-

Table 5.2: Features of RF for MALP.

| Feature name | Data type | Encoding | Null count |
|---|---|---|---|
| Aircraft type | Object | Label encoding | 153 |
| Domestic/International | Object | One-hot encoding | 0 |
| Estimated arrival time | Numerical | | 0 |
| Cruise speed | Numerical | | 506 |
| Straight line distance | Numerical | | 0 |
| Route distance | Numerical | | 228 |
| Estimated flight time | Numerical | | 0 |
| Fuel load | Numerical | | 820 |
| High temperature | Numerical | | 0 |
| Low temperature | Numerical | | 0 |
| Humidity | Numerical | | 0 |
| Barometer | Numerical | | 0 |
| Wind direction | Object | One-hot encoding | 0 |
| Wind speed | Numerical | | 0 |

Table 5.3: Best hyperparameter values of the RF method for MALP.

| Hyperparameters | Search space | Best value |
|---|---|---|
| *n_estimators* | [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000] | 200 |
| *min_samples_split* | [2, 3, 4, 5, 6, 7, 8, 9, 10] | 9 |
| *min_samples_leaf* | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] | 6 |
| *max_depth* | [10, 20, 30, 40, 50, 60, 70, 80, 90, 100] | 20 |

rameter values were determined using a grid search method with 5-fold cross-validation on the training dataset $D^{\text{Training}}$.

After predicting the deviation value $\hat{a}_i$ using the ML methods, we calculate the predicted arrival time $\hat{E}_i$ using the formula $\hat{E}_i = E_i + \hat{a}_i$, where the estimated arrival time $T_i$ is assumed to be known in advance. We further use the estimated arrival time $E_i$ to replace the predicted arrival time $\hat{T}_i$ in Equation (3.2) to (3.4) for calculating the MAE, MSE, and RMSE values as benchmarks. According to the prediction results in Table 5.4, the RF method exhibits a smaller MAE compared to the benchmark. However, the MSE and RMSE values are larger, suggesting that while most prediction errors are small, some individual errors are significantly large.

Table 5.4: Prediction results on the testing dataset for MALP.

| Metrics | Benchmark | RF | Reduction (%) |
|---|---|---|---|
| MAE | 27.00 | 21.63 | 19.81 |
| MSE | 3954.36 | 3196.96 | 19.12 |
| RMSE | 62.88 | 56.54 | 10.05 |

After training, testing and validating the RF method, we utilise the prediction results to more accurately estimate distributions for aircraft arrival times. Let $N$ denote the number of pre-determined decision trees. At this stage, the RF method can provide $N$ possible arrival times for each aircraft. We represent the set of these $N$ possible values, given the input feature vector $f_i$, as $\mathcal{P}(f_i)$. The arrival time vector for set $I$ is written as $\widetilde{\boldsymbol{E}} = \left(\widetilde{E}_1, \widetilde{E}_2, \ldots, \widetilde{E}_{|I|}\right)$. Following previous works (Yang et al., 2024), we employ the Cartesian product $\Phi(\widetilde{\boldsymbol{E}}) = \left\{\mathcal{P}(f_1) \times \mathcal{P}(f_2) \times \cdots \times \mathcal{P}(f_{|I|})\right\}$ to approximate the distribution of $\widetilde{\boldsymbol{E}}$. This product includes $N^{|I|}$ elements, which grows exponentially with $|I|$ (Yan et al., 2022). Accordingly, the uncertain arrival times are represented by a set of scenarios, $\Xi = \{1, 2, \ldots, |\Phi(\widetilde{E})|\}$, with size $N^{|I|}$.

Considering all scenarios in $\Xi$ would make solving the subsequent SP-MIR model overly complex. Thus, we employ the SAA method to approximate the distribution of aircraft arrival times by randomly selecting a subset of scenarios, $\Omega$, from $\Xi$, with each scenario having an equal probability $\frac{1}{|\Omega|}$.

### 5.3.3 Optimisation-enhanced learning-driven scenario generation method

In the prediction results presented in Table 5.4, we observe the following: the MAE of the RF method is 21.63, while the RMSE is 56.54. The MAE indicates the average of the absolute differences between the predicted and actual values, whereas the RMSE emphasises the square of these differences, being more sensitive to larger errors. Consequently, the higher RMSE value suggests the presence of prediction errors. This

indicates that certain decision trees within the RF method may exhibit poor predictive performance, leading to larger prediction errors. This also implies that when the LSG method, based on the RF method, generates scenarios, it may produce some extreme scenarios due to the influence of certain extreme prediction values. In SP approaches, such extreme scenarios can affect the efficiency and robustness of the model. Therefore, it is essential to eliminate these extreme scenarios through appropriate scenario selection methods to enhance the models' performance.

In this section, we propose the OLSG method to mitigate the generation of extreme scenarios. The framework of this method is similar to that of LSG provided in Figure 5.2. The difference is that the OLSG method employs the p-median problem for scenario selection, aiming to minimise the type-1 Wasserstein distance among the selected scenarios while determining an appropriate number of scenarios. In the OLSG method, similar to the LSG method, we first use the RF method to generate the scenario set $\Xi$. The p-median problem is then employed to select $|\Omega|$ scenarios from $\Xi$. However, as noted in Subsection 5.3.2, $\Xi$ is a set containing $N^{|I|}$ scenarios, and directly using it as the input for the p-median problem would render the problem intractable. To ensure the tractability of the p-median problem, we use the SAA method to randomly choose a subset $\Xi^{\text{sub}}$ from $\Xi$. Following this, p-median problem (5.2) is employed to generate the scenario set $\Omega$ from $\Xi^{\text{sub}}$, which is subsequently used in the SP model. In the OLSG method, the set size $|\Omega|$ is defined in advance. The size of the subset $\Xi^{\text{sub}}$ extracted from the original set $\Xi$ is defined as $|\Xi^{\text{sub}}| = \gamma|\Omega|$, where $\gamma$ represents the selection ratio. It should be noted that when $\gamma = 1$, the OLSG method is equivalent to the LSG method, as the scenarios in $\Omega$ are directly selected from $\Xi$ using SAA. As the value of $\gamma$ increases, the scenarios in $\Omega$ need to be selected from a larger set $\Xi^{\text{sub}}$.

In the p-median problem, we minimise the type-1 Wasserstein distance between scenarios selected to the set $\Omega$. The parameter $w^{\xi',\xi}$ denotes the type-1 Wasserstein distance between scenario $\xi'$ and $\xi$. The number of scenarios selected into the set $\Omega$

is constrained by the parameter $|\Omega|$. A binary decision variable $l^{\xi}$ equals 1 if scenario $\xi' \in \Xi^{\text{sub}}$ is incorporated into the scenario set $\Omega$, and 0 otherwise. Another binary decision variable $q^{\xi',\xi}$ equals 1 if scenario $\xi' \in \Xi^{\text{sub}}$ is mapped to scenario $\xi \in \Omega$, and 0 otherwise. The mathematical formulations of the p-median problem for scenario selection are provided as follows:

$$\min \quad \sum_{\xi' \in \Xi^{\text{sub}}} \sum_{\xi \in \Xi^{\text{sub}}} w^{\xi',\xi} q^{\xi',\xi} \tag{5.2a}$$

$$\text{s.t.} \quad \sum_{\xi \in \Xi} l^{\xi} = |\Omega|, \tag{5.2b}$$

$$q^{\xi',\xi} \leq l^{\xi}, \quad \forall \xi' \in \Xi^{\text{sub}}, \forall \xi \in \Xi^{\text{sub}}, \tag{5.2c}$$

$$\sum_{\xi \in \Xi^{\text{sub}}} q^{\xi',\xi} = 1, \quad \xi' \in \Xi^{\text{sub}}, \tag{5.2d}$$

$$l^{\xi} \in \{0,1\}, \quad \forall \xi \in \Xi^{\text{sub}}, \tag{5.2e}$$

$$q^{\xi',\xi} \in \{0,1\}, \quad \xi' \in \Xi^{\text{sub}}, \forall \xi \in \Xi^{\text{sub}}. \tag{5.2f}$$

The objective function (5.2a) minimises the type-1 Wasserstein distance of the selected scenarios. Constraints (5.2b) ensure $|\Omega|$ scenarios are chosen from $\Xi^{\text{sub}}$. Constraints (5.2c) guarantee that all scenarios $\xi' \in \Xi^{\text{sub}}$ are mapped to one of the selected scenarios in $\Omega$. Constraints (5.2d) make sure that each scenario $\xi'$ is assigned to a scenario in $\Omega$. Constraints (5.2e) and (5.2f) define the range of the decision variables. The scenarios selected in the set $\Omega$ have equal probability $\frac{1}{|\Omega|}$.

## 5.4 SBAC method for MALP

Because of the NP-hard nature of the MALP, existing commercial MIP solvers can only handle test instances with a small number of scenarios within a limited CPU time. This section proposes the SBAC method for effectively and efficiently solving the SP-MIR model for MALP. The original SP-MIR model is first decomposed into a master problem and several disaggregated subproblems. The idea of the SBAC method is to search for the master problem solutions around the neighbourhood of a good stability centre point in the branch-and-cut tree to generate strong cuts. Since the aircraft-to-runway assignment decision variables $x$ are binary variables, the stabilisation can be achieved by adding trust region constraints and reverse local branching constraints to the master problem (Baena et al., 2020). The trust region constraints ensure the Hamming distance of the integer solutions to the stability centre $\hat{x}^{sc}$ is within a trust region radius $\kappa$, where $\kappa \geq 1$. In contrast, the reverse local branching constraints avoid the SBAC method repeatedly searching the neighbourhood $\kappa$ of the previous stability centre $\hat{x}^{sc}$, where no better solution can be found. A set $\mathcal{F}$ of pairs $(\hat{x}^{sc}, \kappa')$ is introduced to record regions excluded by the reverse local branching constraints. The master problem with these constraints added is defined as the stabilised master problem. When incorporating the reverse local branching constraints, the stabilised master problem is easier to solve than the master problem due to the reduction of its feasible region. However, it should be noted that due to the function of trust region constraints, the stabilised master problem considers only a portion of the feasible domain of the original problem. Consequently, the optimal solution to the stabilised master problem cannot provide a valid global lower bound for the original problem. Therefore, once an optimal solution to the stabilised master problem is found, the stabilised master problem without trust region constraints should be solved to obtain an effective global lower bound for the original problem.

The implementation framework of the SBAC method is presented in Algorithm (5). We set the lower bound (LB), upper bound (UB), neighbourhood capacity $nc$, and trust region radius $\kappa$ during the initialisation phase. Additionally, we define the set $\mathcal{F}$, cut pool $\mathcal{M}$, and $\mathcal{N}$ as empty sets. By solving Model (5.1) with only one scenario, the resulting first stage solution is defined as the initial stability centre $\hat{x}^{sc}$. We then solve the stabilised master problem (5.6) with set $\mathcal{F}$, cut pools $\mathcal{M}$ and $\mathcal{N}$ using a MIP solver. During the search, three situations will be encountered. The first situation is an integer feasible solution $\hat{x}^{\zeta}$ is found. The disaggregated subproblems (5.3) are then solved with this solution. Subsequently, the resulting Benders cuts (5.4) and (5.5) are added to the unfathomed nodes. Additionally, the Benders cuts (5.4) and (5.5) are incorporated in the cut pools $\mathcal{M}$ and $\mathcal{N}$, respectively. The second situation is an optimal solution $\hat{x}^{so}$ for the stabilised master problem (5.6) is found. The UB is updated with the related optimal solution value, and the stabilised master problem (5.6) without the trust region constraints (5.6c) is solved to obtain the LB. If UB is equal to LB, the SBAC method finds the optimal solution to the original problem and terminates. Otherwise, solve a new stabilised master problem (5.6) with $\mathcal{F}$, $\mathcal{M}$ and $\mathcal{N}$, after the pair $(\hat{x}^{sc}, \kappa)$ is added to $\mathcal{F}$ and the stability centre $\hat{x}^{sc}$ is updated as $\hat{x}^{so}$. The last situation is the stabilised master problem (5.6) is infeasible. If $\kappa \geq nc$, the SBAC method finds the optimal solution to the original problem and terminates. Otherwise, solve a new stabilised master problem (5.6) with $\mathcal{F}$, $\mathcal{M}$ and $\mathcal{N}$, after the pair $(\hat{x}^{sc}, \kappa)$ is added to $\mathcal{F}$ and choose a new trust region radius $\kappa$. The initial value of $\kappa$ can be set to a number greater than 0 and less than or equal to $nc$. When the stabilised master problem (5.6) becomes infeasible, the value of $\kappa$ is increased to a value greater than the current value of $\kappa$ and less than or equal to $nc$. In Subsection 5.6.1, we discuss the impact of various trust region radius updating schemes on the computational performance of the SBAC method.

It is easy to prove that the SBAC method presented in Algorithm (5) can solve the

**Algorithm 5** Stabilised branch-and-check method

1: Initialisation: Set LB $\leftarrow 0$, UB $\leftarrow \infty$, $nc \leftarrow |R| * |I|$, $\kappa \leftarrow \lceil 0.1 * nc \rceil$ , $\mathcal{F} \leftarrow \emptyset$, $\mathcal{M} \leftarrow \emptyset$, $\mathcal{N} \leftarrow \emptyset$.
2: Generate stability centre $\hat{x}^{sc}$.
3: Solve the stabilised master problem (5.6) with $\hat{x}^{sc}$, $\mathcal{F}$, $\mathcal{M}$ and $\mathcal{N}$ using a MIP solver.
4: **If** the stabilised master problem (5.6) is infeasible **then**
5:  **If** $\kappa \geq nc$ **then**
6:  Stop, UB is the optimal value of the original problem.
7:  **End if**
8:  Add reverse local branching constraint $(\hat{x}^{sc}, \kappa)$ to $\mathcal{F}$, choose a new trust region radius $\kappa$, go to line 3.
9: **Else**
10:  **If** an optimal solution $\hat{x}^{so}$ of stabilised master problem (5.6) is found **then**
11:  Obtain optimal objective value $f(\hat{x}^{so})$, update UB $\leftarrow f(\hat{x}^{so})$.
12:  Solve the stabilised master problem (5.3) without trust region constraints (5.6c), obtain optimal objective value $v$, LB $\leftarrow v$.
13:  **If** LB = UB **then**
14:  Stop, UB is the optimal value of the original problem.
15:  **End if**
16:  Add reverse local branching constraint $(\hat{x}^{sc}, \kappa)$ to $\mathcal{F}$, change the stability centre $\hat{x}^{sc} \leftarrow \hat{x}^{so}$, go to line 3.
17:  **Else**
18:  An integer feasible solution $\hat{x}^{\zeta}$ is found.
19:  Solve disaggregated subproblems (5.3) with $\hat{x}^{\zeta}$.
20:  Add Benders cuts (5.4) and (5.5) to the unfathomed nodes.
21:  Add Benders cuts (5.4) and (5.5) to cut pools $\mathcal{M}$ and $\mathcal{N}$, respectively.
22:  **End if**
23: **End if**

optimisation problem within a finite number of iterations:

**Proposition 5.4.1.** *The SBAC method can solve the SP-MIR model for MALP within a finite number of iterations.*

*Proof*: The SBAC method stops if an optimal solution is found. During the search, one of the following actions will be taken: (i) add the Benders optimality cuts to unexplored nodes; (ii) change the stability centre; (iii) increase the trust region radius. The number of Benders cuts is finite. The number of stability centres is finite since the feasible region of the first-stage decision variables $x$ is a combinatorial bounded set. The stability centre will not be searched repeatedly due to the reverse local branching constraints, and Benders cuts added to the stabilised master problem during the search. The values of trust region radius $\kappa$ are limited and never repeated because it is a monotonically increasing sequence bounded by a neighbourhood capacity $nc$. Therefore, the SBAC method will stop within a finite number of iterations and return an optimal solution. □

### 5.4.1 Disaggregated subproblems

The subproblem makes the aircraft sequencing and scheduling decisions for each runway $r \in R$ under each scenario $\omega \in \Omega$, which can be disaggregated into $|R|*|\Omega|$ disconnected subproblems. When the stabilised master problem (5.6) is solved at iteration $\zeta$, we learn the aircraft-to-runway assignment decisions $\hat{x}^{\zeta}$. The notation $\hat{\mathcal{H}}^{r\zeta}$ refers to the clusters of aircraft assigned to runway $r \in R$ by the stabilised master problem solution obtained at iteration $\zeta$, i.e., $\hat{\mathcal{H}}^{r\zeta} = \left\{ i | \hat{x}_i^{r\zeta} = 1, \forall i \in I \right\}$. A disaggregated subproblem for runway $r$ under scenario $\omega$ can be formulated as model (5.3), where the $r$ superscript of $y$ is ignored.

$$\min \; W_1 z^{\omega} + W_2 \left( C^{\text{Fuel}} \sum_{i \in \hat{\mathcal{H}}^{r\zeta}} \alpha_i d_i^{\omega} + C^{\text{CO}_2} \sum_{i \in \hat{\mathcal{H}}^{r\zeta}} \beta \alpha_i d_i^{\omega} + C^{\text{CO}} \sum_{i \in \hat{\mathcal{H}}^{r\zeta}} \varepsilon_i^{\text{CO}} d_i^{\omega} \right.$$

$$\left. + C^{\text{HC}} \sum_{i \in \hat{\mathcal{H}}^{r\zeta}} \varepsilon_i^{\text{HC}} d_i^{\omega} + C^{\text{NO}_x} \sum_{i \in \hat{\mathcal{H}}^{r\zeta}} \varepsilon_i^{\text{NO}_x} d_i^{\omega} + C^{\text{SO}_2} \sum_{i \in \hat{\mathcal{H}}^{r\zeta}} \varepsilon_i^{\text{SO}_2} d_i^{\omega} \right) \tag{5.3a}$$

$$\text{s.t.} \quad \sum_{j \in \hat{\mathcal{H}}^{r\zeta} \cup \{e\} \setminus \{i\}} y_{ij}^{\omega} = 1, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{5.3b}$$

$$\sum_{j \in \hat{\mathcal{H}}^{r\zeta}} y_{sj}^{\omega} = 1, \tag{5.3c}$$

$$\sum_{i \in \hat{\mathcal{H}}^{r\zeta}} y_{ie}^{\omega} = 1, \tag{5.3d}$$

$$\sum_{j \in \hat{\mathcal{H}}^{r\zeta} \cup \{s\} \setminus \{i\}} y_{ji}^{\omega} = \sum_{j \in \hat{\mathcal{H}}^{r\zeta} \cup \{e\} \setminus \{i\}} y_{ij}^{\omega}, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{5.3e}$$

$$t_i^{\omega} \geq E_i^{\omega}, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{5.3f}$$

$$t_i^{\omega} + S_{ij} - t_j^{\omega} \leq M^{\omega}(1 - y_{ij}^{\omega}), \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \forall j \in \hat{\mathcal{H}}^{r\zeta}, i \neq j, \tag{5.3g}$$

$$z^{\omega} \geq t_i^{\omega} - A^{\omega}, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{5.3h}$$

$$d_i^{\omega} \geq t_i^{\omega} - E_i^{\omega}, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{5.3i}$$

$$y_{ij}^{\omega} \in \{0, 1\}, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta} \cup \{s\}, \forall j \in \hat{\mathcal{H}}^{r\zeta} \cup \{e\}, i \neq j, \tag{5.3j}$$

$$t_i^{\omega} \in \mathbb{R}^+, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{5.3k}$$

$$z^{\omega} \in \mathbb{R}^+, \tag{5.3l}$$

$$d_i^{\omega} \in \mathbb{R}^+, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}. \tag{5.3m}$$

### 5.4.2 Benders optimality cuts

After all the disaggregated subproblems are solved, Benders cuts should be added to the unfathomed nodes. Notably, we do not impose a UB on the landing time. This is because arriving aircraft may occasionally experience a prolonged hold before landing, although this is rare. Therefore, all solutions from the first stage are feasible to the second-stage problem, i.e., the SP-MIR model of MALP has complete recourse. Given the complete recourse property of the SP-MIR model for MALP, only optimality cuts are generated during the convergence process of the SBAC method. After all the disaggregated subproblems are solved, we learn the makespan $\hat{\mathcal{Z}}^{r\omega\zeta}$ and the total environmental costs $\hat{\mathcal{C}}^{r\omega\zeta}$ of runway $r$ under scenario $\omega$ at iteration $\zeta$. Then, the related Benders optimality cuts are written as Eq. (5.4) and (5.5). The value of $M$ is set to $10\hat{\mathcal{Z}}^{r\omega\zeta}$ for Eq. (5.4) and $10\hat{\mathcal{C}}^{r\omega\zeta}$ for Eq. (5.5), respectively. In subsequent iterations, cuts (5.4) ensure that if all aircraft in the set $\hat{\mathcal{H}}^{r\zeta}$ are assigned to runway $r$, the makespan under scenario $\omega$ should be greater than or equal to $\hat{\mathcal{Z}}^{r\omega\zeta}$. Similarly, cuts (5.5) guarantee that if all aircraft in the set $\hat{\mathcal{H}}^{r\zeta}$ are assigned to runway $r$, the environmental costs of runway $r$ under scenario $\omega$ should be no less than $\hat{\mathcal{C}}^{r\omega\zeta}$.

$$\eta^{\omega} \geq \hat{\mathcal{Z}}^{r\omega\zeta} - M \sum_{i \in \hat{\mathcal{H}}^{r\zeta}} (1 - x_i^r), \quad \forall r \in R, \forall \omega \in \Omega, \tag{5.4}$$

$$\theta^{r\omega} \geq \hat{\mathcal{C}}^{r\omega\zeta} - M \sum_{i \in \hat{\mathcal{H}}^{r\zeta}} (1 - x_i^r), \quad \forall r \in R, \forall \omega \in \Omega. \tag{5.5}$$

### 5.4.3 Stabilised master problem

The initial stabilised master problem is written as follows:

$$\min \quad \sum_{\omega \in \Omega} p^{\omega} (W_1 \eta^{\omega} + W_2 \sum_{r \in R} \theta^{r\omega}) \tag{5.6a}$$

$$\text{s.t.} \quad \textit{Constraints } (5.1b),\ (5.1k), \tag{5.6b}$$

$$\sum_{(r,i):\hat{x}_i^{r,sc}=1} (1 - x_i^r) + \sum_{(r,i):\hat{x}_i^{r,sc}=0} x_i^r \leq \kappa, \tag{5.6c}$$

$$\sum_{(r,i):\hat{x}_i^{r,sc'}=1} (1 - x_i^r) + \sum_{(r,i):\hat{x}_i^{r,sc'}=0} x_i^r \geq \kappa' + 1, \quad \forall (\hat{x}^{sc'}, \kappa') \in \mathcal{F}, \tag{5.6d}$$

$$\eta^{\omega} \geq \hat{\mathcal{Z}}^{m\omega} - M \sum_{i \in \hat{\mathcal{H}}^m} (1 - x_i^r), \quad \forall m \in \mathcal{M}, \forall r \in R, \forall \omega \in \Omega, \tag{5.6e}$$

$$\theta^{r\omega} \geq \hat{\mathcal{C}}^{n\omega} - M \sum_{i \in \hat{\mathcal{H}}^n} (1 - x_i^r), \quad \forall n \in \mathcal{N}, \forall r \in R, \forall \omega \in \Omega, \tag{5.6f}$$

$$\eta^{\omega} \in \mathbb{R}^+, \quad \forall \omega \in \Omega, \tag{5.6g}$$

$$\theta^{r\omega} \in \mathbb{R}^+, \quad \forall r \in R, \forall \omega \in \Omega. \tag{5.6h}$$

The trust region constraints (5.6c) guarantee that the integer solutions identified in the branch-and-cut tree remain within a trust region radius $\kappa$ of a stability centre $\hat{x}^{sc}$. The reverse local branching constraints (5.6d) are used to avoid the SBAC method repeatedly searching the neighbourhood $\kappa'$ of the previous stability centre $\hat{x}^{sc'}$, where no better solution can be found. The set $\mathcal{F}$ of pairs $(\hat{x}^{sc'}, \kappa')$ record the previous stability centres $\hat{x}^{sc'}$ and related trust region radius $\kappa'$. Constraints (5.6e) and (5.6f) are initial cuts for adding the previously found Benders cuts to the root node of the

new branch-and-cut tree. Constraints (5.6g) and (5.6h) define the domain of decision variables.

## 5.5 Numerical experiments for MALP

This section designs the experiments and discusses specific test instances in detail. Subsequently, we conduct actual sample analysis and implement real-world applications to evaluate the performance of the SP-MIR model for MALP, supported by the OLSG method. We then compare the performance of the SP-MIR and SP-CR models for MALP. Finally, we examine the impact of the decision-maker preference levels.

### 5.5.1 Experimental design for MALP

The experiments are conducted based on real data from HKIA, which is one of the busiest passenger airport and cargo gateway over the world. HKIA has a three-runway system as shown in Figure 5.3, where the northern runway (07L/25R) is used for landings, the central runway (07C/25C) is implemented for take-offs, and the southern runway (07R/25L) adapts its mode for take-offs or landings depending on the hourly aircraft traffic flow.

The test instances are generated using real data from HKIA, including 61 aircraft scheduled to land between 18:00 and 20:00 on 31st October 2023. Given the higher number of approaching aircraft compared to departing ones during this period, it is assumed that both the northern and southern runways are dedicated to landings. Since the arriving aircraft mainly belong to the heavy and large weight classes, we considered only these two classes in our experiments. Following the test instance generation method used by Khassiba et al. (2022), we divided the 61 aircraft into 11 instances. Each instance contains ten aircraft, except for the last instance, which has 11 aircraft. For example, the first instance considers the first ten aircraft, while the second instance

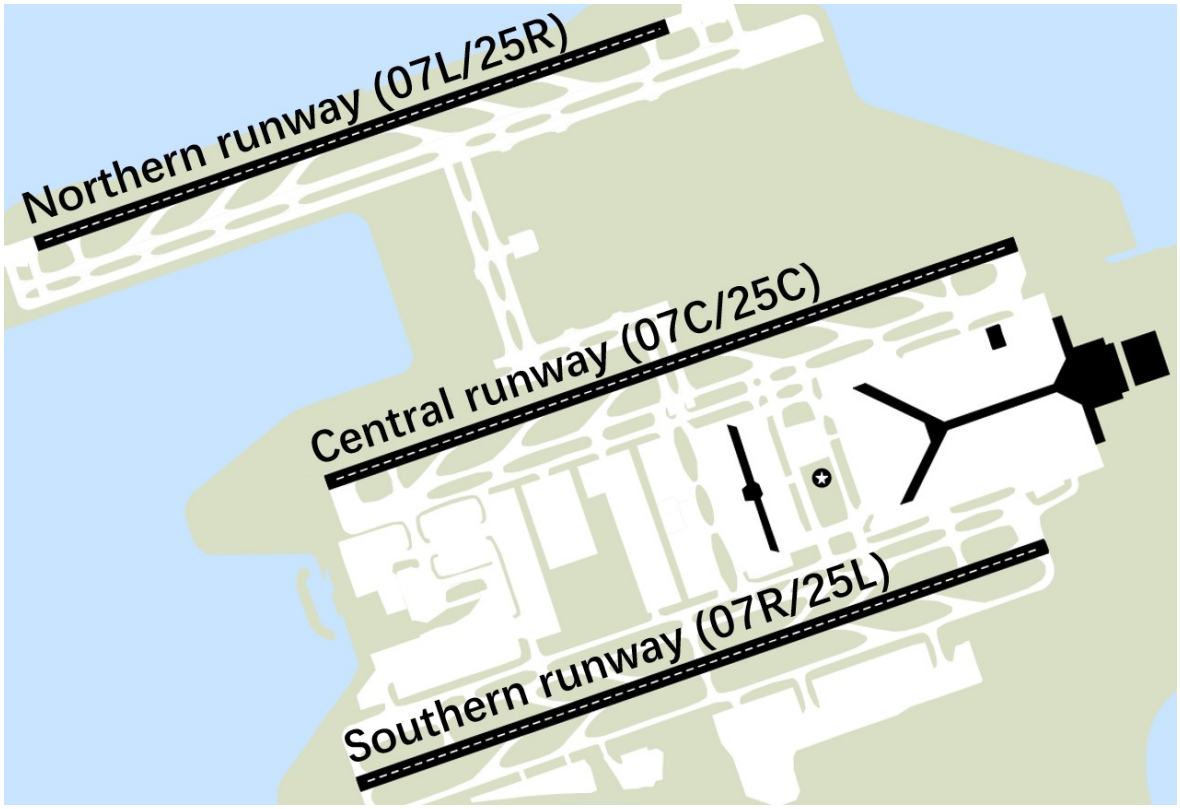


Figure 5.3: Three runway system at HKIA.

includes the 6th to the 15th aircraft. We introduce the information of test instances in Table 5.5.

We sample 10, 50, and 100 independent scenarios for each instance using the scenario generation methods described in Section 5.3. We set an upper limit of 1,200 seconds for the CPU time of the OLSG methods. The CPU times required by the OLSG methods are presented in Figure 5.4, where the LSG method is considered a special case of the OLSG method with a selection ratio $\gamma = 1$. In general, we observe that the OLSG method requires more CPU time as the number of $|\Omega|$ increases. Additionally, as the value of $\gamma$ increases, i.e., when scenarios in $\Omega$ are selected from larger $\Xi^{\text{sub}}$, the OLSG method also demands more CPU time. Specifically, when $\gamma = 1$ in the OLSG method (at this point, the OLSG method is equivalent to the LSG method),

Table 5.5: Detailed information of test instances for MALP.

| Test instances | Aicraft number | Heavy aircraft | Large aircraft |
|---|---|---|---|
| T_1 | 10 | 5 | 5 |
| T_2 | 10 | 4 | 6 |
| T_3 | 10 | 2 | 8 |
| T_4 | 10 | 1 | 9 |
| T_5 | 10 | 1 | 9 |
| T_6 | 10 | 3 | 7 |
| T_7 | 10 | 2 | 8 |
| T_8 | 10 | 1 | 9 |
| T_9 | 10 | 2 | 8 |
| T_10 | 10 | 2 | 8 |
| T_11 | 11 | 1 | 9 |

the CPU times required to generate scenario sets $\Omega$ of different sizes are very small and almost negligible. When $\gamma = 3$, the required CPU times are around 1 second. When $\gamma = 5$, with scenario set sizes $\Omega$ of 10, 50, and 100, the required CPU times are 0.08, 2.30, and 12.70 seconds, respectively, still within an acceptable range. Lastly, when $\gamma = 10$, the required CPU time remains small at 0.70 seconds for a scenario set size $\Omega$ of 10. However, when $\Omega$ is 50 or 100, the required CPU times increase significantly to 568.70 and 1,200.94 seconds, respectively.

Serhan et al. (2018) provided the fuel burn rate for each aircraft type. As this chapter only considers the aircraft weight classes, we summed and averaged the fuel burn rate values separately for aircraft types in the same weight class. The results of fuel burn rates of heavy and large aircraft are 3.39 and 1.32 lb/second, respectively. Following the setting of Sölveling et al. (2011), we provide the emission rates and costs. The emission amount of $CO_2$ is proportional to the fuel flow, and the factor $\beta$ is 3.14, i.e., 1 lb of jet fuel emits 3.14 lb of $CO_2$. Table 5.6 presents the emission rates. The external costs per lb of exhaust emissions $CO_2$, CO, HC, $NO_x$, and $SO_2$ are set as \$0.09, \$0.024, \$3.6, \$4.1, and \$3.9, respectively. Table 5.7 presents the separation time requirements for all aircraft type combinations, following the setting of Pohl et al.

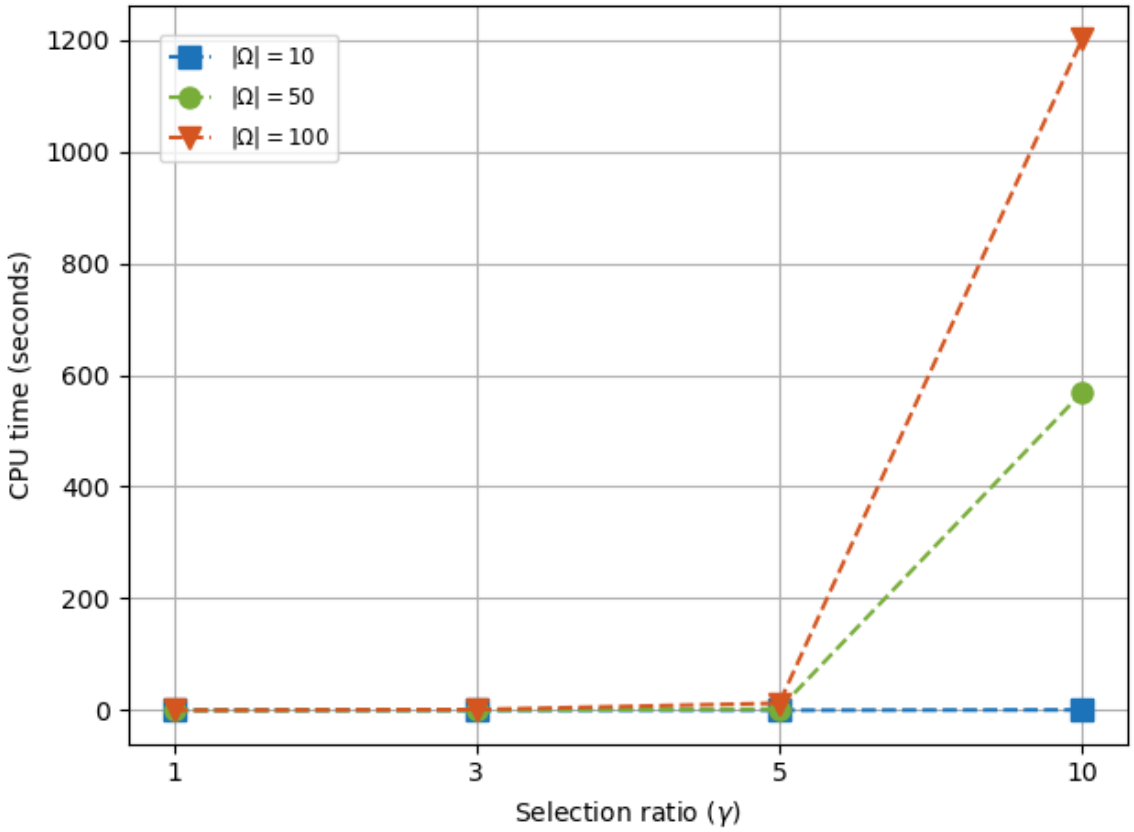


Figure 5.4: CPU time required by the OLSG method.

Table 5.6: Emission rates (in lb/second)

| | CO | HC | $NO_x$ | $SO_2$ |
|---|---|---|---|---|
| Heavy | 0.0041 | 0.0005 | 0.0262 | 0.0021 |
| Large | 0.0023 | 0.0005 | 0.0149 | 0.0010 |

(2021). Given that different selection ratios $\gamma$ are considered in the OLSG method, we use the notation $\text{OLSG}_\gamma$ to represent the OLSG method under various $\gamma$ values.

Table 5.7: Separation time requirements (in seconds)

| Leading | Trailing | |
|---|---|---|
| | Heavy | Large |
| Heavy | 96 | 157 |
| Large | 60 | 69 |

### 5.5.2 Actual sample analysis for MALP

When employing scenario generation methods supported by ML methods to develop multi-runway aircraft landing plans, we strive to ensure that the plans are closely aligned with real-world conditions. Therefore, this subsection will evaluate the effectiveness of plans provided by various optimisation methods in actual scenarios through actual sample analysis. We use $Q(\hat{\mathbf{x}}, \omega)$ to represent the objective function of the second-stage problem, which also serves as the total objective function since the MALP model lacks a first-stage objective function. The term $Q(\hat{\mathbf{x}}, \omega)$ represents the optimal costs under the first-stage decision $\hat{\mathbf{x}}$ and the revealed scenario $\omega$. In the actual sample analysis, we solve the second-stage problem using the optimal solution $\hat{\mathbf{x}}$ with the actual scenario $\omega^{\text{actual}}$, thereby obtaining $Q\left(\hat{\mathbf{x}}, \omega^{\text{actual}}\right)$. In this experiment, both $W_1$ and $W_2$ are set to 0.5.

This experiment compares the deterministic model and the SP-MIR model supported by HDSG, LSG, $\text{OLSG}_3$, $\text{OLSG}_5$, and $\text{OLSG}_{10}$ methods. The overall results, presented in Figure 5.5, illustrate the performance of each optimisation approach in the context of the actual sample analysis. It should be pointed out that since scenarios are not considered in the deterministic model, its performance in actual sample analysis is consistent under different scenario numbers and is only used as a benchmark reference. We find that, in general, scenario generation methods supported by the ML method,

including LSG and OLSG, show a trend of better performance in actual sample analysis as more scenarios are considered in the SP-MIR model they support. This indicates that leveraging historical and auxiliary data can lead to a more accurate estimation of the distribution of uncertain aircraft arrival times, allowing for multi-runway aircraft landing plans provided to be more aligned with actual scenarios. However, in actual sample analysis, the performance of the HDSG method does not improve as the number of scenarios considered by the SP-MIR model increases. Instead, it performs the worst when the number of scenarios reaches 100. This may be because the distribution of uncertain parameters cannot be accurately estimated using historical data alone. Consequently, as the number of scenarios increases, more extreme scenarios are generated due to the inaccurate distribution of uncertain parameters. These extreme scenarios are incorporated into the SP-MIR model, resulting in the formulated plans being unsatisfactory in actual scenarios.

Specifically, when the number of scenarios is 10, the performance of the scenario generation methods, except for the HDSG method, is inferior to the benchmark deterministic model in actual scenarios due to the limited number of scenarios included. The total objective function of the HDSG method is 0.63% lower than that of the deterministic model, while the other methods are approximately 0.57% to 2.46% higher than the benchmark. When the number of scenarios increases to 50, the SP-MIR model supported by all scenario generation methods performs better than the deterministic model in actual scenarios. The total objective function is reduced by approximately 1.31% to 4.45%. As the number of scenarios increases to 100, the SP-MIR model supported by all scenario generation methods involving the ML method performs better than the deterministic model in actual scenarios, with its total objective function reduced by approximately 4.47% to 9.92%. Conversely, the SP-MIR model driven by the HDSG method performs poorly, with its objective function slightly exceeding the benchmark by 0.41%. Additionally, we find that when the LSG method is considered a special

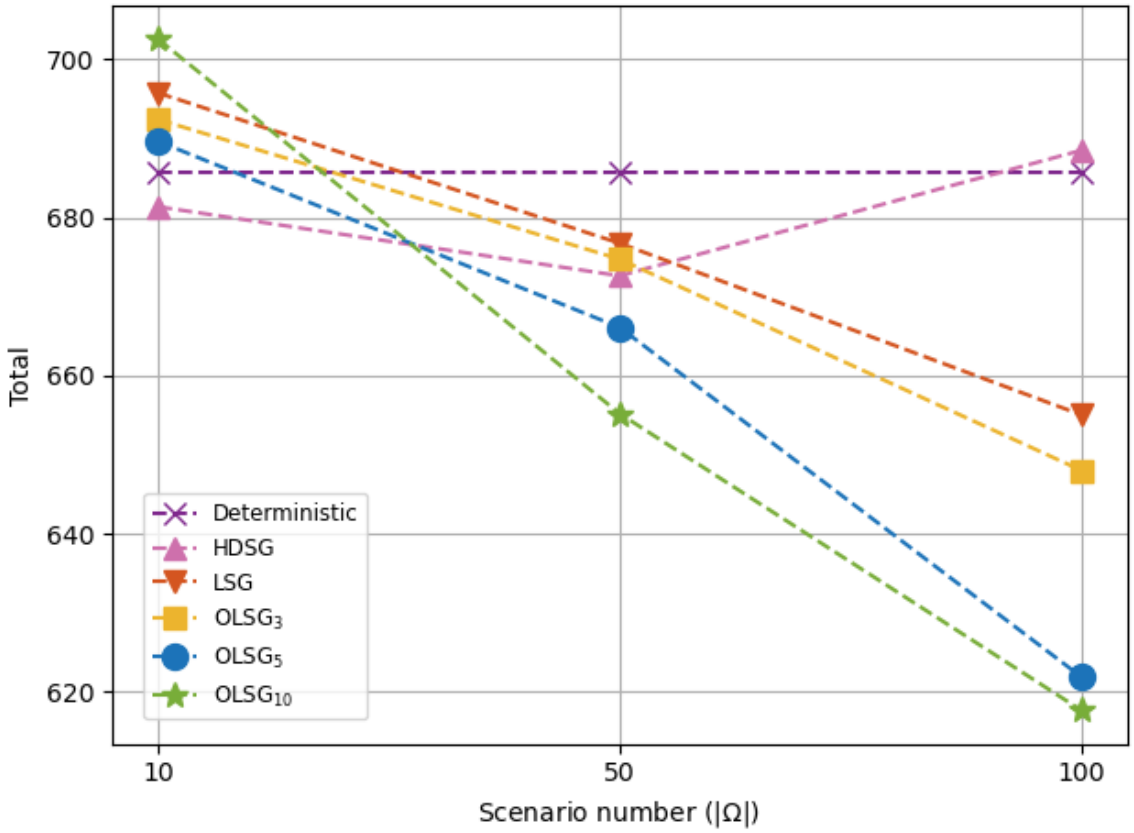


Figure 5.5: Overall results of the actual sample analysis for MALP.

form of the OLSG method with a selection ratio of $\gamma = 1$, the SP-MIR model driven by the OLSG method demonstrates that larger $\gamma$ values lead to better performance at scenario numbers 50 and 100.

We then present the detailed results of the actual sample analysis in Table 5.8, where the notation (Env.) represents the environmental costs. The deterministic model used as the benchmark has a makespan of 1,082.18 and environmental costs of 289.15. We find that the makespan indicator remains relatively stable under various circumstances, fluctuating between approximately 1,070 and 1,080. Environmental costs follow a similar pattern to the previously discussed total objective value. With 10 scenarios, the HDSG method reduces environmental costs by 2.34% compared to

Table 5.8: Detail results of the actual sample analysis for MALP.

| | $\|\Omega\| = 10$ | | $\|\Omega\| = 50$ | | $\|\Omega\| = 100$ | |
|---|---|---|---|---|---|---|
| | Makespan | Env. | Makespan | Env. | Makespan | Env. |
| HDSG | 1,080.27 | 282.39 | 1,084.55 | 260.66 | 1,083.36 | 293.56 |
| LSG | 1,075.82 | 315.62 | 1,075.73 | 277.69 | 1,069.91 | 240.13 |
| $OLSG_3$ | 1,074.73 | 309.99 | 1,082.36 | 267.03 | 1,076.55 | 219.46 |
| $OLSG_5$ | 1,075.73 | 303.48 | 1,077.73 | 254.42 | 1,073.27 | 170.50 |
| $OLSG_{10}$ | 1,072.45 | 332.61 | 1,076.91 | 233.44 | 1,070.27 | 165.08 |

the deterministic model, while the costs for other methods are approximately 4.96% to 15.03% higher than the benchmark. When the number of scenarios increases to 50, the SP-MIR model, leveraging various scenario generation methods, outperforms the deterministic model in actual scenarios, with reductions in environmental costs ranging from 3.96% to 19.26%. Further increasing the scenarios to 100, the SP-MIR model, supported by scenario generation methods involving the ML method, significantly surpasses the deterministic model, reducing environmental costs by 16.96% to 42.90%. In contrast, the SP-MIR model driven by the HDSG method underperforms, with its environmental costs exceeding the benchmark by 1.52%. In addition, for all scenario generation methods supported by the ML method, environmental costs decrease as the number of scenarios increases. When the number of scenarios increases from 10 to 100, the LSG method demonstrates a reduction of 23.91%, the $OLSG_3$ method achieves a reduction of 29.20%, the $OLSG_5$ method results in a reduction of 43.81%, and the $OLSG_{10}$ method exhibits the largest reduction of 50.38% in environmental costs.

Regarding the overall results of the actual sample analysis, the $OLSG_{10}$ method with 100 scenarios performs better than other methods. The $OLSG_5$ method with 100 scenarios is the second best, with a total objective value only 0.68% higher than the best. It is worth noting that generating 100 scenarios using the $OLSG_{10}$ method takes 1,200.94 seconds, which is 98.94% longer than the 12.70 seconds required by the $OLSG_5$ method. Given that the MALP requires near real-time optimisation, the

$OLSG_5$ method with 100 scenarios may be a better choice.

### 5.5.3 Real-world implementation for MALP

In this subsection, we evaluate the performance of the SP models for MALP driven by different scenario generation methods through real-world implementation based on real-world scenarios. The SP models for MALP are designed to operate in a rolling manner, advancing five aircraft at a time (Khassiba et al., 2022). Each test instance generated in Subsection 5.5.1 is treated as a decision period. The last five aircraft in each decision period overlap with the first five aircraft in the next period, except for the final period. This overlapping method prevents short-sighted decision-making. Similar methods have also been employed in other transportation studies (Solak et al., 2018; Shui and Szeto, 2018; Khassiba et al., 2022). Specifically, for each test instance generated in Subsection 5.5.1, except for test instance T_11, the first five aircraft in the optimal solution $\hat{\mathbf{x}}$ are fixed as $\hat{\mathbf{x}}^*$. For the last test instance T_11, $\hat{\mathbf{x}}^*$ is the same as the optimal solution $\hat{\mathbf{x}}$. We then solve the second-stage problem $Q\left(\hat{\mathbf{x}}^*, \omega^{\text{actual}}\right)$, where $\omega^{\text{actual}}$ denotes the actual scenario. Considering that in the actual sample analysis, the HDSG method achieves the best performance with 50 scenarios, while the OLSG methods achieve the best performance with 100 scenarios, we use these settings in real-world implementation. The weights $W_1$ and $W_2$ are also set to 0.5 in this experiment.

According to the results presented in Table 5.9, the HDSG method exhibits the poorest overall performance, with total objective value, makespan, and environmental costs being 436.04, 536.73, and 335.35, respectively. These values are 5.13%, 1.30%, and 11.89% higher than the deterministic model's. This poor performance may be attributed to the HDSG method's reliance solely on historical data to generate scenarios, which may not accurately reflect the true distribution of uncertain parameters. Consequently, the subsequent SP-MIR models fail to provide satisfactory multi-runway aircraft landing plans, occasionally performing worse than the deterministic model in

real-world implementation. The LSG method, which incorporates vast historical and auxiliary data through the ML method to generate scenarios, demonstrates improved performance. Its total objective value, makespan, and environmental costs are 385.23, 521.55, and 248.92, respectively. These values represent reductions of 7.12%, 1.56%, and 16.94% compared to the deterministic model. The OLSG methods with different $\gamma$ values all demonstrate good performance. Among the three, the $\text{OLSG}_3$ method performs the worst in the real-world implementation. The $\text{OLSG}_{10}$ method outperforms the $\text{OLSG}_3$ method, achieving total objective value, makespan, and environmental costs of 334.15, 521.55, and 146.76, respectively. These values are 11.82%, 0.36%, and 37.40% lower than those of the $\text{OLSG}_3$ method. The $\text{OLSG}_5$ method performs the best, with a total objective value, makespan, and environmental costs of 329.72, 524.91, and 134.54, respectively. These values are 12.98%, 0.28%, and 42.62% lower than those of the $\text{OLSG}_3$ method, and 1.33%, $-0.64\%$, and 8.33% lower than those of the $\text{OLSG}_{10}$ method. The superior performance of the OLSG methods can be attributed to using the Wasserstein distance-based optimisation method to select the generated scenarios further. This approach mitigates the negative effects of scenarios that deviate significantly from real-world scenarios due to the limited predictive accuracy of the single decision tree in the RF method. As a result, the generated scenarios more closely resemble the true distribution of uncertain parameters, enabling the subsequent SP-MIR model to provide satisfactory multi-runway aircraft landing plans.

In addition, through the line graphs of cumulative environmental costs (fuel consumption and exhaust emission costs) in Figure 5.6, we observe that the $\text{OLSG}_5$ and $\text{OLSG}_{10}$ methods exhibit significantly lower environmental costs than other methods over time. This demonstrates that the scenario generation methods supported by the ML method, when incorporated into the SP-MIR models, can more effectively capture complex environmental changes and subsequently develop multi-runway aircraft landing plans with substantially lower environmental costs. This advantage becomes

Table 5.9: Results of the real-world implementation for MALP.

| | Total | Makespan | Environmental |
|---|---|---|---|
| Deterministic | 414.77 | 529.82 | 299.72 |
| HDSG | 436.04 | 536.73 | 335.35 |
| LSG | 385.23 | 521.55 | 248.92 |
| $OLSG_3$ | 378.94 | 523.45 | 234.43 |
| $OLSG_5$ | 329.72 | 524.91 | 134.54 |
| $OLSG_{10}$ | 334.15 | 521.55 | 146.76 |

increasingly apparent over time.

### 5.5.4 Comparison of SP-MIR and SP-CR models for MALP

In this subsection, we compare the performance of the SP-MIR and SP-CR models through real-world implementation. As mentioned in the introduction, both models adopt a two-stage decision process. The main difference is that aircraft-to-runway assignments and sequencing decisions are made in the first stage of SP-CR, while the second stage focuses on aircraft scheduling decisions. Given that advanced aviation technologies provide accurate aircraft arrival times when ATC makes sequencing decisions. Therefore, SP-MIR positions aircraft sequencing decisions in the second stage, where the uncertain aircraft arrival times are revealed. We provide the SP-CR model for MALP as follows:

$$\min \sum_{\omega\in\Omega} p^{\omega}\left[W_1 z^{\omega} + W_2\left(C^{\text{Fuel}}\sum_{i\in I}\alpha_i d_i^{\omega} + C^{\text{CO}_2}\sum_{i\in I}\beta\alpha_i d_i^{\omega} + C^{\text{CO}}\sum_{i\in I}\varepsilon_i^{\text{CO}} d_i^{\omega}\right.\right.$$

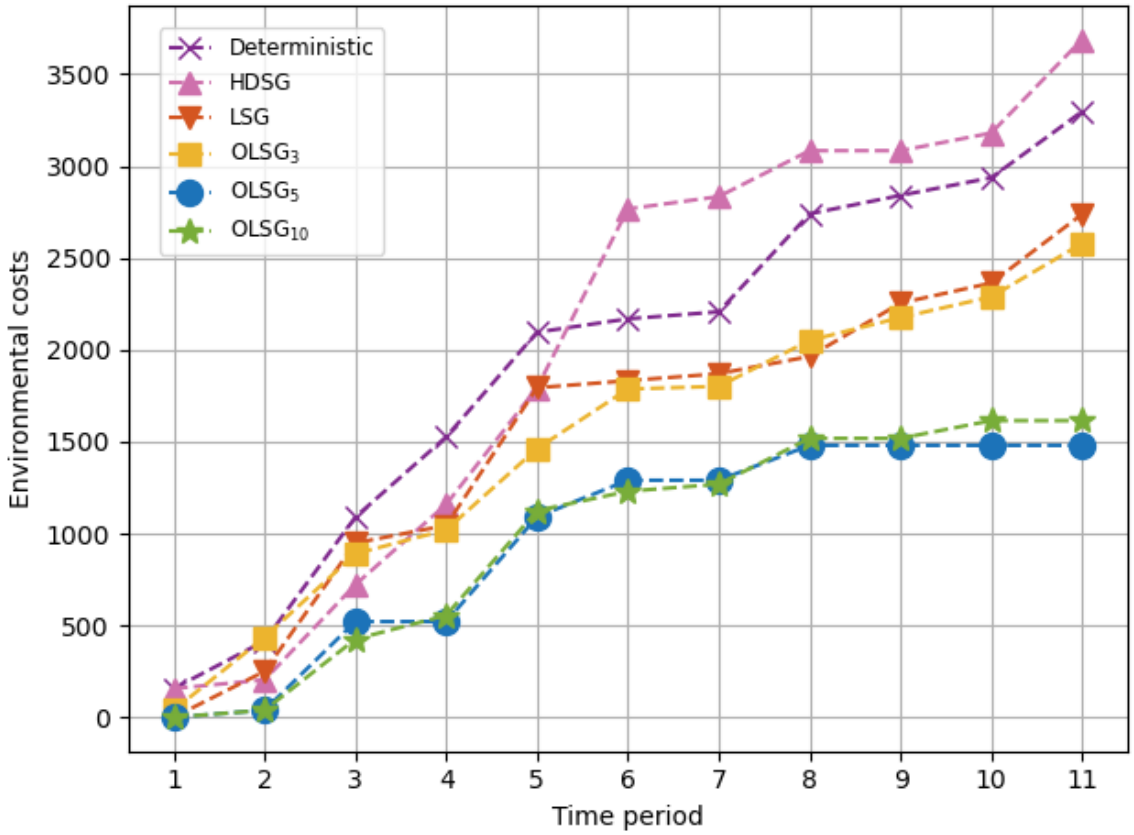


Figure 5.6: Cumulative environmental costs for MALP.

$$+C^{\mathrm{HC}}\sum_{i\in I}\varepsilon_i^{\mathrm{HC}}d_i^{\omega}+C^{\mathrm{NO}_x}\sum_{i\in I}\varepsilon_i^{\mathrm{NO}_x}d_i^{\omega}+C^{\mathrm{SO}_2}\sum_{i\in I}\varepsilon_i^{\mathrm{SO}_2}d_i^{\omega}\Big)\Big] \tag{5.7a}$$

$$\text{s.t.}\quad \textit{Constraints } (5.1b),\ (5.1g),\ (5.1k),\ (5.1m)-(5.1o), \tag{5.7b}$$

$$x_i^r=\sum_{j\in I\cup\{e\}\setminus\{i\}}y_{ij}^r,\quad \forall r\in R,\forall i\in I, \tag{5.7c}$$

$$\sum_{j\in I}y_{sj}^r=1,\quad \forall r\in R, \tag{5.7d}$$

$$\sum_{i\in I}y_{ie}^r=1,\quad \forall r\in R, \tag{5.7e}$$

$$\sum_{j \in I \cup \{s\} \setminus \{i\}} y_{ji}^{r} = \sum_{j \in I \cup \{e\} \setminus \{i\}} y_{ij}^{r}, \quad \forall r \in R, \forall i \in I, \tag{5.7f}$$

$$t_i^{\omega} + S_{ij} - t_j^{\omega} \leq M^{\omega}(1 - y_{ij}^{r}), \quad \forall r \in R, \forall i \in I, \forall j \in I, i \neq j, \forall \omega \in \Omega, \tag{5.7g}$$

$$y_{ij}^{r} \in \{0, 1\}, \quad \forall r \in R, \forall i \in I \cup \{s\}, \forall j \in I \cup \{e\}, i \neq j. \tag{5.7h}$$

Given that the scenarios generated by the $\text{OLSG}_5$ method performed the best in the previous experimental results, we use 100 scenarios generated by the $\text{OLSG}_5$ method when comparing the two SP models. The weights $W_1$ and $W_2$ are both set as 0.5. The average total objective value, makespan, and environmental costs of the SP-CR model over 11 time periods are 445.47, 539.73, and 351.21, respectively. In comparison, the SP-MIR model achieves 329.72, 524.91, and 134.54 for these metrics. When compared to the SP-CR model, the SP-MIR model's performance is reduced by 25.98%, 2.75%, and 61.68% in the three indicators, respectively. In terms of overall performance, the SP-MIR model reduced the total objective value by 25.98% compared to the SP-CR model. This indicates that utilising the SP-MIR model for decision-making in a multi-runway system could lead to higher operational efficiency and better resource conservation. Additionally, the SP-MIR model demonstrated a significant reduction in environmental costs (61.68%), underscoring its superior performance in minimising environmental impact. Consequently, adopting the SP-MIR model can enable more sustainable and environmentally friendly airport runway operations, which is particularly important for airports aiming to fulfil their social responsibility and reduce their environmental footprint. Furthermore, the SP-MIR model demonstrated a slight improvement in makespan (reduced by 2.75%), which, although modest, indicates it has certain advantages in airport runway efficiency. This is particularly critical for runway operations that require strict time control. Additionally, the results for each time period are presented in Figure 5.7. It can be observed that, for most time periods,

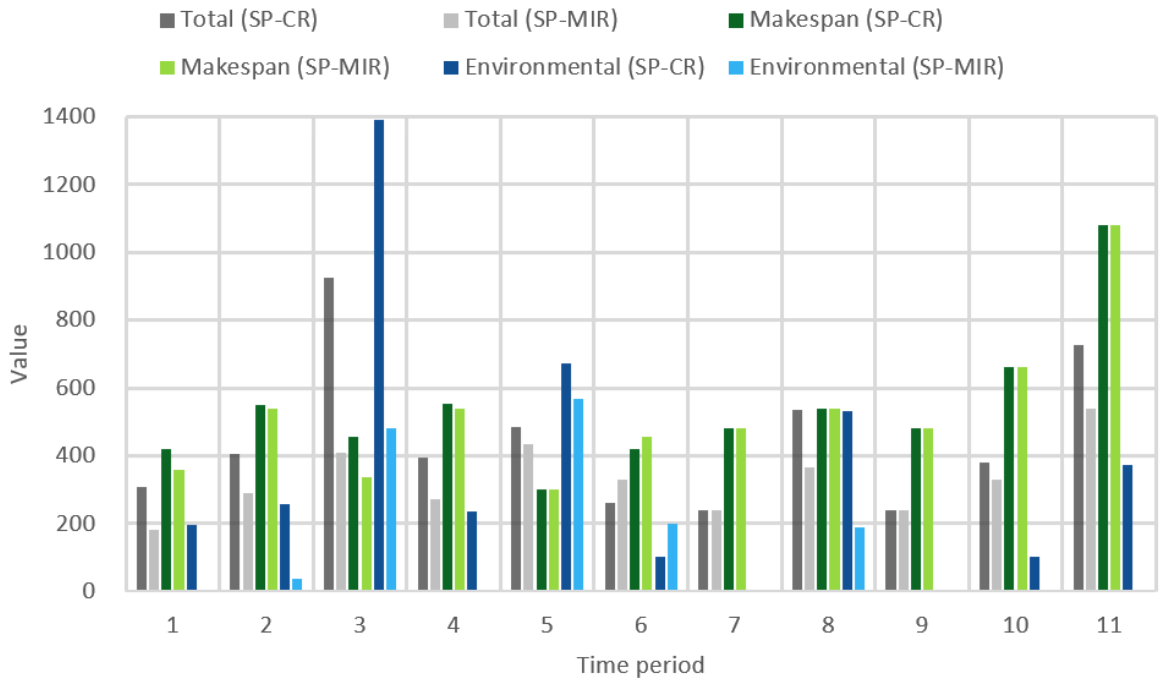


Figure 5.7: Comparison results of SP-MIR and SP-CR models for MALP.

the SP-MIR model outperforms the SP-CR model across all indicators, particularly in terms of environmental costs. This superior performance is mainly due to the SP-MIR model considering aircraft sequencing decisions in the second stage, which allows it to handle aircraft sequencing and scheduling decisions more flexibly after the uncertain arrival times are revealed, thereby outperforming the SP-CR model in real-world implementation.

### 5.5.5 The impact of the decision-maker preference levels

The weights $W_1$ and $W_2$ are used to reflect the decision-makers' preferences for makespan and environmental costs, respectively, where $W_1 + W_2 = 1$. The total objective value, makespan, and environmental costs of optimal decisions under different preference levels are provided in Figure 5.8. As $W_1$ decreases and $W_2$ increases (i.e., decision-makers prioritise environmental factors), makespan gradually increases, indicating a slight de-

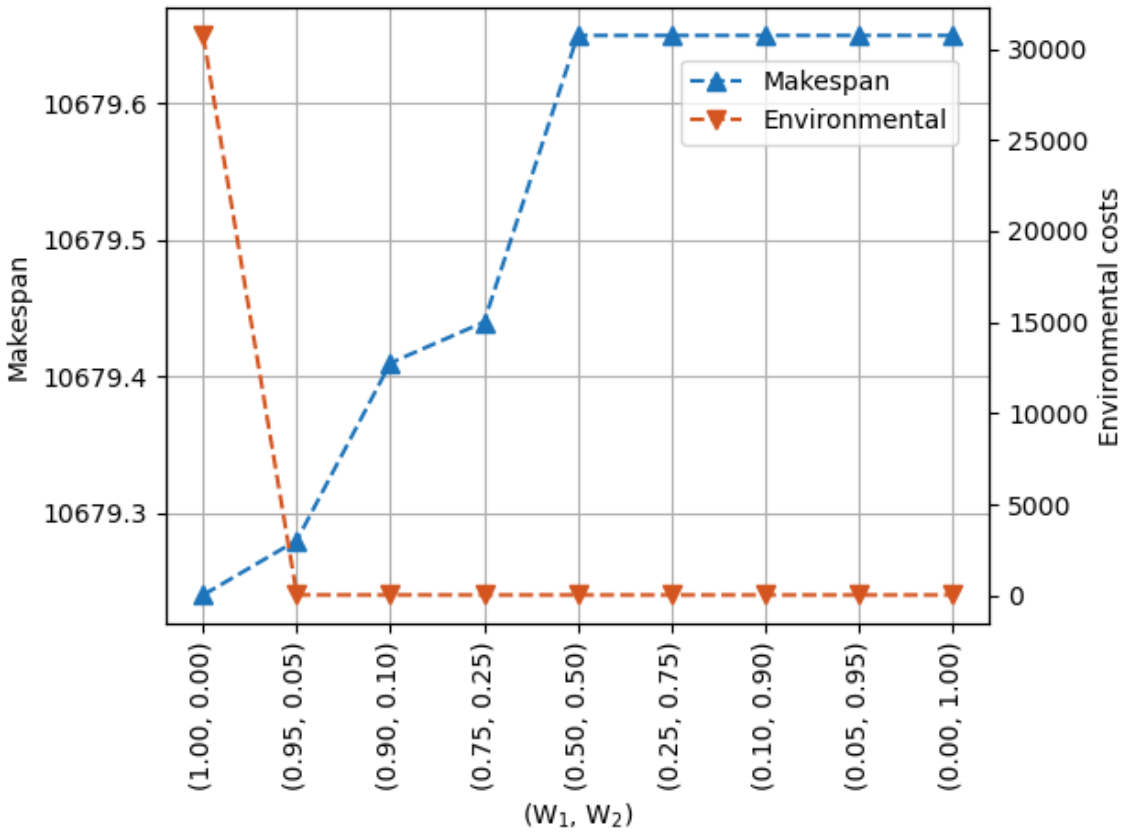


Figure 5.8: The impact of preference levels on makespan and environmental costs.

crease in operational efficiency. For the environmental costs, prioritising environmental factors can initially lead to substantial reductions in fuel consumption costs and emissions expenses. Subsequently, with environmental preference weight increases, these cost reductions tend to be more gradual. In summary, decision-makers face a trade-off between operational efficiency (makespan) and environmental costs (fuel consumption and exhaust emission costs). They must achieve a delicate balance between efficiency and environmental impact. By making informed choices, they can achieve sustainable outcomes while maintaining operational effectiveness.

### 5.5.6 Managerial implications and insights

The above numerical experiments yield the following managerial implications and insights for ATC.

(i) The findings underscore the importance of integrating scenario generation methods supported by ML, such as LSG and OLSG, within the SP-MIR model for MALP. These methods significantly enhance performance in actual scenarios by reducing fuel consumption and exhaust emissions while maintaining operational efficiency. ATC is encouraged to adopt these models to drive the SP-MIR model for MALP, achieving effective and environmentally friendly multi-runway aircraft landing plans in real-world implementation.

(ii) The results highlight the advantages of adopting the SP-MIR model in multi-runway aircraft landing operations over the SP-CR model used in previous studies. The SP-MIR model significantly reduces the total objective value, thereby enhancing operational efficiency and resource conservation in a multi-runway system, which is crucial for optimising airport operations. Additionally, by adopting the SP-MIR model, airports can significantly reduce environmental costs, thereby contributing to more sustainable and environmentally friendly operations, aligning with social responsibility goals and reducing the environmental footprint. Although the SP-MIR model shows a modest improvement in the makespan indicator, this is particularly critical for runway operations requiring strict time control. Overall, the SP-MIR model performs well across all indicators and is recommended for integration into the planning process by decision-makers. By leveraging the capabilities of the SP-MIR model, airports can develop efficient and environmentally friendly multi-runway aircraft landing plans.

(iii) The findings emphasise the need for ATC to balance operational efficiency and environmental impact when planning aircraft landings. By prioritising environmental factors, ATC can substantially reduce fuel consumption and exhaust emissions, lead-

ing to cost savings and environmental benefits, though this may slightly increase the makespan. Therefore, decision-makers are encouraged to adopt strategies that balance efficiency with environmental responsibility to achieve sustainable and efficient multi-runway aircraft landing plans.

## 5.6 Scalability analyses for MALP

The scalability analyses assess the performance of the proposed SBAC method. All solution methods are implemented in Python using the commercial solver GUROBI. The experiments take place on a computer featuring an Intel Core i7-12700K CPU (12 cores, 20 threads) at 5.00 GHz, along with 32 GB of memory. Solution methods terminate when the CPU time reaches 1200 seconds. In the scalability analyses, we set the preference for makespan $W_1 = 0.5$ and preference for environmental costs $W_2 = 0.5$. The scenarios of each test instance are generated by the $\text{OLSG}_5$ method. In Subsection 5.6.1, we evaluate the impact of trust region radius updating schemes on the SBAC method. In Subsection 5.6.2, we compare the computational performance of the proposed SBAC method with benchmark solution methods.

### 5.6.1 The impact of trust region radius updating schemes on the SBAC method

In this subsection, we examine the impact of trust region radius updating schemes on the SBAC method. First, we introduce several trust region radius updating schemes, starting with the scheme proposed by Fischetti and Lodi (2003). In this scheme, if the stabilised master problem is infeasible, the radius value $\kappa$ is updated to $\kappa \leftarrow \kappa + \lceil \frac{\kappa}{2} \rceil$, where the initial value of $\kappa$ is set to $\lceil 0.1 \times nc \rceil$. The second scheme follows Baena et al. (2020). This scheme initially searches the solution space within a small radius, then expands to a medium radius, and finally covers the entire solution space. In

this scheme, the initial value of $\kappa$ is set to $\lceil 0.1 \times nc \rceil$. After the stabilised master problem becomes infeasible during the convergence process, the value of $\kappa$ increases successively, taking values of $\kappa \in \{\lceil 0.2 \times nc \rceil, \lceil 0.5 \times nc \rceil, nc\}$. Finally, we use a new trust region radius updating scheme that initially searches the solution space within a medium radius, subsequently expanding to cover the entire solution space. In this scheme, the initial value of $\kappa$ is set to $\lceil 0.5 \times nc \rceil$. When the stabilised master problem becomes infeasible, $\kappa$ is immediately updated to $nc$. For simplicity, the three schemes mentioned above are denoted as Scheme 1, Scheme 2, and Scheme 3, respectively.

Since all three schemes driving the SBAC method achieved optimal solutions for all test instances, we report only the CPU time indicator in Table 5.10. We find that Scheme 1 tends to have the highest CPU time, indicating it is the least efficient of the three schemes. Scheme 2 performs better, generally being the second most efficient. Scheme 3 consistently shows the lowest average CPU time across most test instances, making it the most efficient in terms of CPU time. Overall, the CPU time for Scheme 3 is 13.79% less than that for Scheme 1 and 10.82% less than that for Scheme 2. For all schemes, the CPU time increases with $|\Omega|$, as larger sets of scenarios typically require more computing resources. As $|\Omega|$ increases, the efficiency gap between the schemes becomes more pronounced. For instance, when $|\Omega| = 10$, Scheme 3 saves 10.76% and 9.03% of computing time compared to Schemes 1 and 2, respectively. When $|\Omega| = 100$, the performance advantage of Scheme 3 is even more apparent, saving 12.78% and 9.83% of computing time compared to Schemes 1 and 2, respectively. In conclusion, Scheme 3 not only has the best average performance but also demonstrates consistent performance across various scenario sizes, making it a reliable choice for handling the SP model for MALP. In our subsequent comparisons, Scheme 3 is adopted as the default trust region radius updating scheme for the SBAC method.

Table 5.10: Comparison of CPU time for different trust region radius updating schemes in the SBAC method

| | Scheme 1 | Scheme 2 | Scheme 3 |
|---|---|---|---|
| $\|\Omega\| = 10$ | | | |
| T_1 | 1.57 | 1.52 | **1.30** |
| T_2 | 2.01 | 2.03 | **1.99** |
| T_3 | 1.64 | 1.55 | **1.45** |
| T_4 | 1.81 | 1.81 | **1.27** |
| T_5 | **1.79** | **1.79** | 2.36 |
| T_6 | 1.27 | 1.19 | **0.91** |
| T_7 | 1.25 | 1.25 | **0.98** |
| T_8 | 1.28 | 1.23 | **1.14** |
| T_9 | 1.26 | 1.20 | **0.91** |
| T_10 | 1.37 | 1.33 | **1.28** |
| T_11 | 2.10 | 2.12 | **1.97** |
| Average | 1.58 | 1.55 | **1.41** |
| $\|\Omega\| = 50$ | | | |
| T_1 | 9.88 | 10.30 | **7.01** |
| T_2 | 8.79 | 8.26 | **6.25** |
| T_3 | 11.17 | 10.64 | **10.61** |
| T_4 | **10.52** | 10.96 | 12.83 |
| T_5 | 9.83 | 9.12 | **7.36** |
| T_6 | 9.31 | 8.59 | **7.49** |
| T_7 | 10.58 | 9.98 | **6.89** |
| T_8 | 7.55 | 7.31 | **6.61** |
| T_9 | 6.83 | 6.50 | **5.64** |
| T_10 | 8.71 | 8.15 | **7.57** |
| T_11 | 14.06 | 13.54 | **10.83** |
| Average | 9.75 | 9.40 | **8.10** |
| $\|\Omega\| = 100$ | | | |
| T_1 | 23.66 | 22.57 | **16.68** |
| T_2 | 29.32 | 28.42 | **24.63** |
| T_3 | 21.05 | 21.36 | **17.35** |
| T_4 | 36.18 | 35.94 | **34.84** |
| T_5 | 29.17 | **27.65** | 37.59 |
| T_6 | 16.17 | 14.93 | **13.15** |
| T_7 | 25.27 | 24.48 | **10.57** |
| T_8 | 20.19 | **19.32** | 32.92 |
| T_9 | 15.18 | 14.55 | **12.18** |
| T_10 | 24.22 | 23.16 | **14.93** |
| T_11 | 41.84 | 40.62 | **31.35** |
| Average | 25.66 | 24.82 | **22.38** |
| Overall | 12.33 | 11.92 | **10.63** |

### 5.6.2 Comparison of the SBAC method and the benchmark solution methods

We utilise GUROBI and the previously proposed LBBD and BAC methods (Elçi and Hooker, 2022; Guo and Zhu, 2023; Li et al., 2023) as benchmark methods to evaluate the performance of the SBAC method by comparing the optimality gap and CPU time metrics.

Table 5.11 shows the results, where "Gap (%)" indicates the optimality gap, and "Time (s)" indicates the required CPU time. The methods that performed the best in terms of CPU time for the test instances in Table 5.11 are highlighted in bold. Overall, the SBAC method demonstrated superior performance, achieving the optimal solution for all test instances with an overall CPU time of only 10.63 seconds. This represents a reduction of 84.56%, 98.72%, and 89.79% in overall CPU time compared to the GUROBI, LBBD, and BAC methods, respectively. Specifically, SBAC consistently demonstrates the shortest CPU time for most test instances. Its performance remains excellent across a wide range of $|\Omega|$ values and significantly outperforms other solution methods, particularly when $|\Omega| = 100$. SBAC's consistency and stability across different situations make it a reliable choice for the SP model of the MALP under various sizes. While GUROBI performs well when $|\Omega| = 10$, its CPU time increases significantly with larger sizes, potentially affecting its scalability in practical applications. Among the remaining two solution methods, the LBBD method consistently exhibits high gaps (%) and very long CPU times, particularly when $|\Omega|$ values are large, resulting in the worst overall performance. Although the BAC method finds optimal solutions for all test instances, its CPU time requirements are substantial.

Subsequently, we compare the SBAC method and the benchmark methods in terms of CPU time metrics using the Wilcoxon signed-rank test. As shown in Table 5.12, all the P values are less than 0.01, indicating a highly significant reduction in CPU time

Table 5.11: Comparison of computational performance for different solution methods for MALP

| | GUROBI | | LBBD | | BAC | | SBAC | |
|---|---|---|---|---|---|---|---|---|
| | Gap (%) | Time (s) | Gap (%) | Time (s) | Gap (%) | Time (s) | Gap (%) | Time (s) |
| $\|\Omega\| = 10$ | | | | | | | | |
| T_1 | 0.00 | **0.85** | 0.00 | 92.21 | 0.00 | 13.81 | 0.00 | 1.30 |
| T_2 | 0.00 | **1.41** | 0.00 | 11.74 | 0.00 | 78.41 | 0.00 | 1.99 |
| T_3 | 0.00 | 1.67 | 0.00 | 111.48 | 0.00 | 11.65 | 0.00 | **1.45** |
| T_4 | 0.00 | 1.61 | 0.00 | 210.87 | 0.00 | 16.27 | 0.00 | **1.27** |
| T_5 | 0.00 | **1.71** | 0.00 | 134.39 | 0.00 | 16.39 | 0.00 | 2.36 |
| T_6 | 0.00 | **0.73** | 0.00 | 66.91 | 0.00 | 8.60 | 0.00 | 0.91 |
| T_7 | 0.00 | 1.11 | 0.00 | 113.23 | 0.00 | 15.77 | 0.00 | **0.98** |
| T_8 | 0.00 | **1.10** | 0.00 | 55.81 | 0.00 | 10.81 | 0.00 | 1.14 |
| T_9 | 0.00 | 1.08 | 0.00 | 56.79 | 0.00 | 8.45 | 0.00 | **0.91** |
| T_10 | 0.00 | **1.25** | 0.00 | 100.31 | 0.00 | 17.54 | 0.00 | 1.28 |
| T_11 | 0.00 | **1.80** | 0.00 | 586.83 | 0.00 | 30.58 | 0.00 | 1.97 |
| Average | 0.00 | **1.30** | 0.00 | 140.05 | 0.00 | 20.75 | 0.00 | 1.41 |
| $\|\Omega\| = 50$ | | | | | | | | |
| T_1 | 0.00 | 30.15 | 1.86 | 1,203.80 | 0.00 | 78.90 | 0.00 | **7.01** |
| T_2 | 0.00 | 15.76 | 0.00 | 1,182.03 | 0.00 | 90.37 | 0.00 | **6.25** |
| T_3 | 0.00 | 25.47 | 2.07 | 1,201.09 | 0.00 | 91.45 | 0.00 | **10.61** |
| T_4 | 0.00 | 50.48 | 2.17 | 1,201.40 | 0.00 | 81.61 | 0.00 | **12.83** |
| T_5 | 0.00 | 41.08 | 0.91 | 1,206.10 | 0.00 | 69.56 | 0.00 | **7.36** |
| T_6 | 0.00 | 20.07 | 1.78 | 1,200.32 | 0.00 | 73.70 | 0.00 | **7.49** |
| T_7 | 0.00 | 16.84 | 2.05 | 1,201.18 | 0.00 | 81.18 | 0.00 | **6.89** |
| T_8 | 0.00 | 15.71 | 0.00 | 1,027.38 | 0.00 | 72.05 | 0.00 | **6.61** |
| T_9 | 0.00 | 12.74 | 0.00 | 758.90 | 0.00 | 67.08 | 0.00 | **5.64** |
| T_10 | 0.00 | 11.64 | 0.00 | 1,082.35 | 0.00 | 84.65 | 0.00 | **7.57** |
| T_11 | 0.00 | 168.42 | 19.27 | 1,208.63 | 0.00 | 192.43 | 0.00 | **10.83** |
| Average | 0.00 | 37.12 | 2.74 | 1,133.40 | 0.00 | 89.36 | 0.00 | **8.10** |
| $\|\Omega\| = 100$ | | | | | | | | |
| T_1 | 0.00 | 156.34 | 5.26 | 1,204.97 | 0.00 | 195.91 | 0.00 | **16.68** |
| T_2 | 0.00 | 101.56 | 5.74 | 1,212.48 | 0.00 | 160.52 | 0.00 | **24.63** |
| T_3 | 0.00 | 56.85 | 3.84 | 1,211.36 | 0.00 | 148.16 | 0.00 | **17.35** |
| T_4 | 0.00 | 346.24 | 5.99 | 1,211.06 | 0.00 | 223.70 | 0.00 | **38.84** |
| T_5 | 0.00 | 249.93 | 21.96 | 1,203.46 | 0.00 | 176.98 | 0.00 | **37.59** |
| T_6 | 0.00 | 98.23 | 15.24 | 1,205.53 | 0.00 | 166.23 | 0.00 | **13.15** |
| T_7 | 0.00 | 59.57 | 3.69 | 1,208.96 | 0.00 | 170.10 | 0.00 | **10.57** |
| T_8 | 0.00 | 148.43 | 17.62 | 1,211.62 | 0.00 | 173.43 | 0.00 | **32.92** |
| T_9 | 0.00 | 27.98 | 5.10 | 1,205.44 | 0.00 | 168.12 | 0.00 | **12.18** |
| T_10 | 0.00 | 132.22 | 5.86 | 1,206.22 | 0.00 | 181.83 | 0.00 | **14.93** |
| T_11 | 0.00 | 471.46 | 29.85 | 1,216.27 | 0.00 | 461.67 | 0.00 | **31.35** |
| Average | 0.00 | 168.07 | 10.94 | 1,208.85 | 0.00 | 202.42 | 0.00 | **22.38** |
| Overall | 0.00 | 68.83 | 4.56 | 827.43 | 0.00 | 104.18 | 0.00 | **10.63** |

Table 5.12: Comparison in CPU time indicator by the Wilcoxon signed-rank test

| Methods | Z score | P value | R value | Strength of effect size |
|---|---|---|---|---|
| GUROBI | -2.251 | 0.000 | 0.392 | Medium effect |
| LBBD | -6.842 | 0.000 | 1.191 | Large effect |
| BAC | -5.842 | 0.000 | 1.017 | Large effect |

achieved by the SBAC method. R value is used to describe the effect size of the SBAC method that ignores the sample size (Cohen, 2013). R value is 0.5 for a large effect, 0.3 for a medium effect, and 0.1 for a small effect (Coolican, 2017). According to the R value, the SBAC method substantially reduces CPU time compared to the LBBD and BAC methods, and moderately reduces CPU time compared to the GUROBI method. Overall, the results of the Wilcoxon signed-rank tests show that the proposed SBAC method has significant statistical improvement in the CPU time indicator compared with the benchmark methods.

## 5.7 Conclusions

This chapter investigates the SP-MIR model for MALP under aircraft arrival time uncertainty. This model for MALP employs a two-stage decision process, wherein arriving aircraft are assigned to runways in the first stage, and the landing sequence and times are scheduled in the second stage. The objective of the SP-MIR model is to ensure efficient and environmentally friendly aircraft landing operations. We employ ML-driven scenario generation methods to create a potentially small finite scenario set. An optimisation-enhanced version is further proposed to generate scenarios that closely reflect actual scenarios based on the estimated distributions, thereby preventing subsequent SP-MIR models from being adversely affected by extreme sce-

narios and avoiding suboptimal decisions. The numerical results demonstrate that integrating advanced machine learning-supported scenario generation methods, such as LSG and OLSG, within the SP-MIR model can effectively formulate efficient and sustainable multi-runway aircraft landing plans in practical applications. Furthermore, compared to the SP-CR model used in previous studies, our proposed SP-MIR model significantly reduces fuel consumption and exhaust emissions while maintaining operational efficiency, thereby enhancing its performance in practical scenarios. Furthermore, numerical studies have shown that runway efficiency objectives may conflict with environment-related objectives. Due to the NP-hardness of the SP model for MALP, we propose a novel SBAC method. This approach decomposes the original problem into an assignment master problem and several sequencing and scheduling subproblems. Trust region constraints and reverse local branching constraints are added to the master problem, stabilising it around a good stability centre point and enabling the generation of strong Benders cuts. We conduct extensive scalability analyses using real data from HKIA. The results show that the SBAC method significantly reduces the CPU time indicator compared to the commercial solver and well-known benchmark methods from the literature.

In the OLSG method proposed in this chapter, the objective of scenario selection is to minimise the type-1 Wasserstein distance between scenarios, but this overlooks the optimisation problems' inherent information. In future research, we will enhance the current OLSG method by incorporating the problems' information to define the proximity measure between scenarios. By comprehensively considering the information of the optimisation problem, we can avoid overfitting, which solely relies on data, generating a more robust set of scenarios. This approach enhances the model's stability in the face of uncertainty, thereby improving its efficiency and effectiveness. In addition, this study focuses on providing efficient and environmentally friendly aircraft landing plans for a multi-runway system, considering only the landing operations. In the future,

we can jointly optimise landing operations and traffic flow management in the terminal airspace. This integrated approach helps prevent suboptimal plans caused by isolated optimisation, improves overall operational efficiency, and reduces environmental impact in the multi-runway system, thereby better meeting the growing demand for air traffic.

# Chapter 6

# Conclusions and future research directions

## 6.1 Conclusions

This thesis investigates prescriptive analytics for airside operations under uncertainties. The first study investigates the AGAP, which aims to optimise aircraft-to-gate assignments, aircraft sequencing for each contact gate, and scheduling plans to minimise apron assignments and delay times. Practical aircraft assignment requirements are considered. Two prescriptive analytics approaches using KNN and RF methods are developed to address the uncertainty in aircraft arrival times. The ETO approach generally yields superior airport gate assignment decisions but requires generating numerous scenarios based on the estimated distributions, leading to scalability issues. The CSR method is used to select a reasonable scenario set for the subsequent SP model, and the BBC method is used to handle the ETO approach efficiently. Numerical experiments using real-world data from XMN Airport demonstrate the effectiveness of these methods. The BBC method exhibits significant computational performance improvements over the commercial solver. Furthermore, we show that the ETO ap-

proach, supported by high-performance ML methods and effective scenario selection strategies, outperforms other optimisation approaches in airport gate assignment plans, proving its effectiveness in real-world applications.

The second study introduces the ETO and ETDRO approaches for the ASSP, demonstrating their effectiveness in enhancing runway operations through the integration of prediction and optimisation techniques. The novel ETDRO approach incorporates unavoidable prediction and sampling errors into decision-making. This approach uses the DRO model to account for situations that deviate from the estimated distribution but are still possible, thus generating more robust decisions. Extensive numerical experiments with real-world data from XMN airport highlight that the ETDRO approach, especially with smaller $\epsilon$ values, provides superior performance in aircraft sequencing and scheduling compared to other optimisation methods. The proposed inexact S-DR BBC method for the ETDRO approach significantly improves computational performance, enabling runway controllers to utilise the ETDRO approach in handling real-world runway operations.

The third study examines the prescriptive analytics of the MALP under aircraft arrival time uncertainty, aiming to design efficient and environmentally friendly aircraft landing operations. Adopting the ETO approach in prescriptive analysis, we employ ML techniques to estimate the distribution of uncertain arrival times. An OLSG method creates scenarios that closely reflect actual scenarios based on these estimated distributions. This approach prevents subsequent SP models from being adversely affected by extreme scenarios, thereby avoiding suboptimal decisions. Experimental results highlight the superior performance of the ETO approach, supported by the OLSG method, over other optimisation approaches. Additionally, we introduce a novel exact SBAC method to solve the ETO approach for the MALP efficiently. Computational experiments demonstrate that the proposed SBAC method achieves statistically significant improvements in CPU time compared to benchmark methods.

This thesis demonstrates the robustness and effectiveness of combining ML techniques, optimisation approaches, uncertainty modelling, and advanced decomposition methods to tackle key operational challenges in airside operations, resulting in significant operational efficiency and economic benefits. Moreover, the successful application of prescriptive analytics underscores the significant potential and advantages of data-driven decision-making in complex aviation operational environments. This suggests that these data-driven decision-making approaches can be introduced as innovative solutions to various operational challenges within the aviation industry.

## 6.2 Future research directions

Several future research directions related to the aforementioned studies are outlined below:

The first research direction is the expansion of datasets and ML techniques. Future research should focus on expanding dataset sizes to improve the accuracy and reliability of machine learning models. By collecting and utilising larger datasets, it will be possible to develop more sophisticated ML techniques capable of handling complex structures and providing precise predictions and estimated distributions. Additionally, exploring advanced ML algorithms such as deep learning, ensemble methods, and hybrid models can further enhance the predictive accuracy and robustness of prescriptive analytics approaches. These improvements will enable more accurate and reliable inputs for subsequent optimisation methods, ultimately leading to better decision-making in airside operations. Enhanced datasets and advanced ML techniques can also support the development of real-time decision-making systems that can adapt to dynamic changes in operational conditions, such as sudden weather changes or unexpected delays. Furthermore, larger datasets can facilitate training models that incorporate diverse and rare events, thereby improving the overall robustness of predictive analytics.

Integrating these advanced methods will enhance operational efficiency, reduce environmental impacts, and promote sustainability in aviation operations.

The second research direction integrates normative analysis with advanced uncertainty modelling and optimisation approaches, such as DRO methods and risk-averse criteria. The distribution of uncertain parameters provided by ML methods inherently contains prediction errors. To mitigate the impact of these errors, combining the ETO approach with DRO can result in more reliable and robust decisions under uncertain conditions. This study combines ETO with a DRO model based on a type-1 Wasserstein ambiguity set, achieving promising results. Future research could explore integrating the ETO method with DRO models using different ambiguity sets. As the performance of the DRO model largely depends on the choice of ambiguity set, it is crucial to select different sets flexibly according to the specific optimisation problems and the nature of prediction errors. This ensures that prediction errors are reasonably incorporated into decision-making, providing efficient and robust decisions. Additionally, when the ETO method generates scenarios, extreme scenarios may arise from certain extreme prediction results. If not properly managed, these extreme scenarios can hinder the optimisation process from yielding satisfactory decisions. Future studies might consider incorporating Conditional Value at Risk (CVaR) and mean-CVaR criteria into the ETO approach. This would allow decision-makers to better account for and mitigate the impact of rare but severe scenarios, balancing expectations with extreme risks. As a result, they can make more reliable and effective decisions that provide stable returns while effectively controlling risks.

The third research direction is the application of prescriptive analytics to other operations within the aviation industry. This thesis demonstrates the effectiveness of prescriptive analytics approaches for the AGAP, ASSP, and MALP through extensive numerical experiments, highlighting their capability to generate high-performance airside operation plans. Future research should focus on extending these prescriptive

analytics approaches to other airside operations and airport decision-making problems. By integrating operations research techniques with machine learning methods, airside controllers can significantly enhance operational efficiency, increase capacity, alleviate congestion, manage disruptions, and maintain high service levels. Comparative studies should be conducted to evaluate the performance of prescriptive analytics across various airside operational contexts, identifying best practices and areas for improvement. This approach can lead to a more holistic optimisation of airport operations, promoting a seamless and efficient airside environment.

Moreover, prescriptive analytics can be expanded to address more macro-level issues within the aviation sector. For instance, airline recovery operations can benefit from optimised scheduling and resource allocation to minimise the impact of disruptions and enhance recovery times. Aviation network design can be improved by using prescriptive analytics to optimise route planning, fleet allocation, and hub selection, thereby increasing the overall efficiency and resilience of the aviation network. Additionally, flight crew scheduling can be optimised to ensure compliance with regulatory requirements while maximising crew utilisation and satisfaction. Maintenance schedules can also be enhanced by predicting and preemptively addressing potential issues, thereby reducing aircraft downtime and improving safety. Expanding the scope of prescriptive analytics to encompass these broader operational areas not only fosters a more integrated and efficient aviation system but also helps address complex, dynamic challenges within the industry. This comprehensive approach to optimisation can ultimately lead to significant operational improvements, cost savings, and enhanced passenger experiences across the entire aviation industry.

# References


Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867. 23, 41, 42, 73, 92

Agra, A. and Rodrigues, F. (2022). Distributionally robust optimization for the berth allocation problem under uncertainty. *Transportation Research Part B: Methodological*, 164:1–24. 72, 73, 83, 87, 92

Baena, D., Castro, J., and Frangioni, A. (2020). Stabilized benders methods for large-scale combinatorial optimization, with application to data privacy. *Management Science*, 66(7):3051–3068. 74, 131, 145, 168

Balakrishnan, H. and Chandran, B. G. (2010). Algorithms for scheduling runway operations under constrained position shifting. *Operations Research*, 58(6):1650–1665. 9, 67, 68, 74, 93, 119, 125

Bansal, M., Huang, K.-L., and Mehrotra, S. (2018). Decomposition algorithms for two-stage distributionally robust mixed binary programs. *SIAM Journal on Optimization*, 28(3):2360–2383. 72, 83, 85, 86, 87

Beck, J. C. (2010). Checking-up on branch-and-check. In *Principles and Practice of Constraint Programming–CP 2010: 16th International Conference, CP 2010, St.*

*Andrews, Scotland, September 6-10, 2010. Proceedings 16*, pages 84–98. Springer. 131

Benlic, U., Burke, E. K., and Woodward, J. R. (2017). Breakout local search for the multi-objective gate allocation problem. *Computers & Operations Research*, 78:80–93. 8, 18

Bennell, J. A., Mesgarpour, M., and Potts, C. N. (2011). Airport runway scheduling. *4OR*, 9:115–138. 1, 2, 9, 13, 68, 73, 125, 126, 130

Bennell, J A and Mesgarpour, Mohammad and Potts, Chris N (2017). Dynamic scheduling of aircraft landings. *European Journal of Operational Research*, 258(1):315–327. 126

Bertsimas, D. and Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044. 3, 13, 14, 19, 21, 70, 71, 129

Bertsimas, D., Sim, M., and Zhang, M. (2019). Adaptive distributionally robust optimization. *Management Science*, 65(2):604–618. 11

Bi, J., Wang, F., Ding, C., Xie, D., and Zhao, X. (2022). The airport gate assignment problem: A branch-and-price approach for improving utilization of jetways. *Computers & Industrial Engineering*, 164:107878. 1, 8, 17, 25, 26

Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media. 60, 79, 102

Black, B., Ainslie, R., Dokka, T., and Kirkbride, C. (2023). Distributionally robust resource planning under binomial demand intakes. *European Journal of Operational Research*, 306(1):227–242. 72

Bolat, A. (2001). Models and a genetic algorithm for static aircraft-gate assignment problem. *Journal of the Operational Research Society*, 52(10):1107–1120. 18

Bouras, A., Ghaleb, M. A., Suryahatmaja, U. S., and Salem, A. M. (2014). The airport gate assignment problem: a survey. *The Scientific World Journal*, 2014(1):923859. 1, 2, 13, 17

Brandt, T., Dlugosch, O., Abdelwahed, A., van den Berg, P. L., and Neumann, D. (2022). Prescriptive analytics in urban policing operations. *Manufacturing & Service Operations Management*, 24(5):2463–2480. 13

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32. 21, 129

Brownlee, A. E., Weiszer, M., Chen, J., Ravizza, S., Woodward, J. R., and Burke, E. K. (2018). A fuzzy approach to addressing uncertainty in airport ground movement optimisation. *Transportation Research Part C: Emerging Technologies*, 92:150–175. 2

Castaing, J., Mukherjee, I., Cohn, A., Hurwitz, L., Nguyen, A., and Müller, J. J. (2016). Reducing airport gate blockage in passenger aviation: Models and analysis. *Computers & Operations Research*, 65:189–199. 8, 18

Chen, S., Wu, L., Ng, K., Liu, W., and Wang, K. (2024). How airports enhance the environmental sustainability of operations: A critical review from the perspective of operations research. *Transportation Research Part E: Logistics and Transportation Review*, 183:103440. 1, 17, 22, 73, 125, 126

Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. Routledge. 173

Coolican, H. (2017). *Research methods and statistics in psychology*. Psychology Press. 173

Daş, G. S. (2017). New multi objective models for the gate assignment problem. *Computers & Industrial Engineering*, 109:347–356. 8

Daş, G. S., Gzara, F., and Stützle, T. (2020). A review on airport gate assignment problems: Single versus multi objective approaches. *Omega*, 92:102146. 1, 2, 7, 8, 13, 17, 18, 26

Delage, E. and Ye, Y. (2010). Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612. 11

Dell'Orco, M., Marinelli, M., and Altieri, M. G. (2017). Solving the gate assignment problem through the fuzzy bee colony optimization. *Transportation Research Part C: Emerging Technologies*, 80:424–438. 8, 26

Deng, W., Li, K., and Zhao, H. (2024). A flight arrival time prediction method based on cluster clustering-based modular with deep neural network. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):6238–6247. 30, 33

Deng, W., Zhao, H., Yang, X., Xiong, J., Sun, M., and Li, B. (2017). Study on an improved adaptive pso algorithm for solving multi-objective gate assignment. *Applied Soft Computing*, 59:288–302. 8, 18

Dorndorf, U., Jaehn, F., and Pesch, E. (2012). Flight gate scheduling with respect to a reference schedule. *Annals of Operations Research*, 194(1):177–187. 18

Dorndorf, U., Jaehn, F., and Pesch, v. (2008). Modelling robust flight-gate scheduling as a clique partitioning problem. *Transportation Science*, 42(3):292–301. 8

Dorndorf, Ulrich and Jaehn, Florian and Pesch, Erwin (2017). Flight gate assignment and recovery strategies with stochastic arrival and departure times. *OR Spectrum*, 39:65–93. 18

Elçi, Ö. and Hooker, J. (2022). Stochastic planning and scheduling with logic-based

benders decomposition. *INFORMS Journal on Computing*, 34(5):2428–2442. 131, 171

Elmachtoub, A. N. and Grigas, P. (2022). Smart "predict, then optimize". *Management Science*, 68(1):9–26. 13

Fachini, R. F. and Armentano, V. A. (2020). Logic-based benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. *Computers & Industrial Engineering*, 148:106641. 131

Fischetti, M. and Lodi, A. (2003). Local branching. *Mathematical programming*, 98:23–47. 168

Galli, L., Levato, T., Schoen, F., and Tigli, L. (2021). Prescriptive analytics for inventory management in health care. *Journal of the Operational Research Society*, 72(10):2211–2224. 21, 71

Gangammanavar, H. and Bansal, M. (2022). Stochastic decomposition method for two-stage distributionally robust linear optimization. *SIAM Journal on Optimization*, 32(3):1901–1930. 72, 73

Gendron, B., Scutellà, M. G., Garroppo, R. G., Nencioni, G., and Tavanti, L. (2016). A branch-and-benders-cut method for nonlinear power design in green wireless local area networks. *European Journal of Operational Research*, 255(1):151–162. 23, 73, 94

Ghoniem, A., Sherali, H. D., and Baik, H. (2014). Enhanced models for a mixed arrival-departure aircraft sequencing problem. *INFORMS Journal on Computing*, 26(3):514–530. 67

Gong, H. and Zhang, Z.-H. (2022). Benders decomposition for the distributionally ro-

bust optimization of pricing and reverse logistics network design in remanufacturing systems. *European Journal of Operational Research*, 297(2):496–510. 73, 74, 131

Guépet, J., Briant, O., Gayon, J.-P., and Acuna-Agost, R. (2016). The aircraft ground routing problem: Analysis of industry punctuality indicators in a sustainable perspective. *European Journal of Operational Research*, 248(3):827–839. 2

Guevara, E., Babonneau, F., Homem-de Mello, T., and Moret, S. (2020). A machine learning and distributionally robust optimization framework for strategic energy planning under uncertainty. *Applied energy*, 271:115005. 72

Guo, P. and Zhu, J. (2023). Capacity reservation for humanitarian relief: A logic-based benders decomposition method with subgradient cut. *European Journal of Operational Research*, 311(3):942–970. 131, 171

Harikiopoulo, D. and Neogi, N. (2010). Polynomial-time feasibility condition for multi-class aircraft sequencing on a single-runway airport. *IEEE transactions on intelligent transportation systems*, 12(1):2–14. 9, 68, 126

Heidt, A., Helmke, H., Kapolke, M., Liers, F., and Martin, A. (2016). Robust runway scheduling under uncertain conditions. *Journal of Air Transport Management*, 56:28–37. 126

Hong, Y., Choi, B., and Kim, Y. (2018). Two-stage stochastic programming based on particle swarm optimization for aircraft sequencing and scheduling. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1365–1377. 127

Hong, Y., Choi, B., Lee, K., and Kim, Y. (2017). Dynamic robust sequencing and scheduling under uncertainty for the point merge system in terminal airspace. *IEEE Transactions on Intelligent Transportation Systems*, 19(9):2933–2943. 126

Hooker, J. N. (2007). Planning and scheduling by logic-based benders decomposition. *Operations Research*, 55(3):588–602. 131

Ikli, S., Mancel, C., Mongeau, M., Olive, X., and Rachelson, E. (2021). The aircraft runway scheduling problem: A survey. *Computers & Operations Research*, 132:105336. 1, 2, 9, 13, 67, 68, 73, 125, 126

Jiang, Y., Wang, Y., Hu, Z., Xue, Q., and Yu, B. (2023). Airport gate assignment problem with harbor constraints based on branch-and-price algorithm. *Transportation Research Part E: Logistics and Transportation Review*, 176:103192. 17

Jiang, Y., Wang, Y., Xiao, Y., Xue, Q., Shan, W., and Zhang, H. (2024). Joint runway–gate assignment based on the branch-and-price algorithm. *Transportation Research Part C: Emerging Technologies*, 162:104605. 17

Kandula, S., Krishnamoorthy, S., and Roy, D. (2021). A prescriptive analytics framework for efficient e-commerce order delivery. *Decision Support Systems*, 147:113584. 3

Kapolke, M., Fürstenau, N., Heidt, A., Liers, F., Mittendorf, M., and Weiß, C. (2016). Pre-tactical optimization of runway utilization under uncertainty. *Journal of Air Transport Management*, 56:48–56. 126, 127

Karsu, Ö., Azizoğlu, M., and Alanlı, K. (2021). Exact and heuristic solution approaches for the airport gate assignment problem. *Omega*, 103:102422. 7, 8, 17, 22, 26

Karsu, Ö. and Solyalı, O. (2023). A new formulation and an effective matheuristic for the airport gate assignment problem. *Computers & Operations Research*, 151:106073. 7, 17

Keutchayan, J., Ortmann, J., and Rei, W. (2023). Problem-driven scenario clustering in stochastic optimization. *Computational Management Science*, 20(1):13. 14, 19

Khassiba, A., Bastin, F., Cafieri, S., Gendron, B., and Mongeau, M. (2020). Two-stage stochastic mixed-integer programming with chance constraints for extended aircraft arrival management. *Transportation Science*, 54(4):897–919. 1, 2, 10, 11, 12, 18, 30, 67, 68, 69, 70, 71, 127, 128, 129, 130

Khassiba, A., Cafieri, S., Bastin, F., Mongeau, M., and 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. 2, 10, 11, 18, 30, 68, 69, 70, 71, 126, 127, 129, 152, 160

Kim, J., Goo, B., Roh, Y., Lee, C., and Lee, K. (2023a). A branch-and-price approach for airport gate assignment problem with chance constraints. *Transportation Research Part B: Methodological*, 168:1–26. 2, 8, 9, 10, 11, 18, 19, 51, 68

Kim, M., Park, T., Jeong, J., and Kim, H. (2023b). Stochastic optimization of home energy management system using clustered quantile scenario reduction. *Applied Energy*, 349:121555. 22, 39

Kleywegt, A. J., Shapiro, A., and Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502. 39

Laporte, G. and Louveaux, F. V. (1993). The integer l-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142. 131

Li, J., Li, K., Tian, Q., and Jin, X. (2024). A column generation-based algorithm for gate assignment problem with combinational gates. *Expert Systems with Applications*, 238:121792. 17

Li, Y., Clarke, J.-P., and 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. 7, 8, 17, 22, 26, 44

Li, Y., Côté, J.-F., Coelho, L. C., Zhang, C., and Zhang, S. (2023). Order assignment and scheduling under processing and distribution time uncertainty. *European Journal of Operational Research*, 305(1):148–163. 131, 171

Lieder, A., Briskorn, D., and Stolletz, R. (2015). A dynamic programming approach for the aircraft landing problem with aircraft classes. *European Journal of Operational Research*, 243(1):61–69. 125, 126, 127

Liu, Z., Wu, Z., Ji, Y., Qu, S., and Raza, H. (2021). Two-stage distributionally robust mixed-integer optimization model for three-level location–allocation problems under uncertain environment. *Physica A: Statistical Mechanics and its Applications*, 572:125872. 72

Liu, Z. and 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. 8, 17, 18

Luo, X., Yan, R., and Wang, S. (2023). Comparison of deterministic and ensemble weather forecasts on ship sailing speed optimization. *Transportation Research Part D: Transport and Environment*, 121:103801. 3, 13, 14, 20

Maheo, A., Kilby, P., and Van Hentenryck, P. (2019). Benders decomposition for the design of a hub and shuttle public transit system. *Transportation Science*, 53(1):77–88. 23, 42

Messaoud, M. B. (2021). A thorough review of aircraft landing operation from practical and theoretical standpoints at an airport which may include a single or multiple runways. *Applied Soft Computing*, 98:106853. 2, 9, 12, 68, 125, 126, 127

Mohajerin Esfahani, P. and Kuhn, D. (2018). Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166. 72, 83

Ng, K., Chen, C.-H., and Lee, C. (2021). Mathematical programming formulations for robust airside terminal traffic flow optimisation problem. *Computers & Industrial Engineering*, 154:107119. 91

Ng, K., Lee, C., Chan, F., Chen, C.-H., and Qin, Y. (2020). A two-stage robust optimisation for terminal traffic flow problem. *Applied Soft Computing*, 89:106048. 126

Ng, K., Lee, C., Chan, F., and Qin, Y. (2017). Robust aircraft sequencing and scheduling problem with arrival/departure delay using the min-max regret approach. *Transportation Research Part E: Logistics and Transportation Review*, 106:115–136. 1, 2, 9, 10, 11, 18, 30, 67, 68, 69, 70, 126

Nikolić, M., Rakas, J., and Teodorović, D. (2024). Formulation of the airport collaborative gate allocation problem and the bee colony optimization solution approach. *Engineering Applications of Artificial Intelligence*, 128:107433. 17

Notz, P. M. and Pibernik, R. (2022). Prescriptive analytics for flexible capacity management. *Management Science*, 68(3):1756–1775. 13

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883. 21

Pohl, M., Artigues, C., and Kolisch, R. (2022). Solving the time-discrete winter runway scheduling problem: A column generation and constraint programming approach. *European Journal of Operational Research*, 299(2):674–689. 9, 68, 96

Pohl, M., Kolisch, R., and Schiffer, M. (2021). Runway scheduling during winter operations. *Omega*, 102:102325. 9, 46, 68, 71, 96, 154

Prakash, R., Desai, J., and Piplani, R. (2022). An optimal data-splitting algorithm for aircraft sequencing on a single runway. *Annals of Operations Research*, pages 1–24. 9, 68, 74, 119

Prakash, R., Piplani, R., and Desai, J. (2021). An optimal data-splitting algorithm for aircraft sequencing on two runways. *Transportation Research Part C: Emerging Technologies*, 132:103403. 9, 68, 74, 119

Qi, M. and Shen, Z.-J. (2022). Integrating prediction/estimation and optimization with applications in operations management. In *Tutorials in operations research: emerging and impactful topics in operations*, pages 36–58. INFORMS. 3, 13, 14, 19, 20, 70

Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2018). Accelerating the benders decomposition method: Application to stochastic network design problems. *SIAM Journal on Optimization*, 28(1):875–903. 22, 23, 41, 73, 92, 130

Rahmaniani, Ragheb and Crainic, Teodor Gabriel and Gendreau, Michel and Rei, Walter (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817. 23, 41, 73, 91, 92

Reese, J. (2006). Solution methods for the p-median problem: An annotated bibliography. *NETWORKS: an international Journal*, 48(3):125–142. 130

Rei, W., Cordeau, J.-F., Gendreau, M., and Soriano, P. (2009). Accelerating benders decomposition by local branching. *INFORMS Journal on Computing*, 21(2):333–345. 23, 73, 94

Ribeiro, N. A., Jacquillat, A., and Antunes, A. P. (2019). A large-scale neighborhood search approach to airport slot allocation. *Transportation Science*, 53(6):1772–1797. 1, 17, 67

Sabar, N. R. and Kendall, G. (2015). An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem. *Omega*, 56:88–98. 125, 126, 127

Salehipour, A. (2020). An algorithm for single-and multiple-runway aircraft landing problem. *Mathematics and Computers in Simulation*, 175:179–191. 126, 127

Sama, M., D' Ariano, A., Corman, F., and Pacciarelli, D. (2017). Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas. *Transportation Research Part C: Emerging Technologies*, 80:485–511. 9, 68

Şeker, M. and Noyan, N. (2012). Stochastic optimization models for the airport gate assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(2):438–459. 2, 8, 11, 18, 19, 22, 30, 51

Serhan, D., Lee, H., and Yoon, S. W. (2018). Minimizing airline and passenger delay cost in airport surface and terminal airspace operations. *Journal of Air Transport Management*, 73:120–133. 154

She, Y., Zhao, Q., Guo, R., and Yu, X. (2022). A robust strategy to address the airport gate assignment problem considering operators' preferences. *Computers & Industrial Engineering*, 168:108100. 18

Shehadeh, K. S. (2023). Distributionally robust optimization approaches for a stochastic mobile facility fleet sizing, routing, and scheduling problem. *Transportation Science*, 57(1):197–229. 23, 42, 72, 73, 83, 92

Shui, C. S. and Szeto, W. (2018). Dynamic green bike repositioning problem–a hybrid rolling horizon artificial bee colony algorithm approach. *Transportation Research Part D: Transport and Environment*, 60:119–136. 160

Solak, S., Solveling, G., Clarke, J.-P. B., and Johnson, E. L. (2018). Stochastic runway scheduling. *Transportation Science*, 52(4):917–940. 1, 2, 9, 10, 11, 12, 18, 67, 68, 69, 70, 71, 96, 105, 125, 126, 127, 128, 129, 137, 138, 160

Solveling, G., Solak, S., Clarke, J.-P., and Johnson, E. (2011). Runway operations optimization in the presence of uncertainties. *Journal of Guidance, Control, and Dynamics*, 34(5):1373–1382. 2, 10, 11, 68, 70, 105, 126, 129

Sölveling, G., Solak, S., Clarke, J.-P. B., and Johnson, E. L. (2011). Scheduling of runway operations for reduced environmental impact. *Transportation Research Part D: Transport and Environment*, 16(2):110–120. 126, 134, 137, 138, 154

Thorsteinsson, E. S. (2001). Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In *Principles and Practice of Constraint Programming—CP 2001: 7th International Conference, CP 2001 Paphos, Cyprus, November 26–December 1, 2001 Proceedings 7*, pages 16–30. Springer. 131

Tian, X., Yan, R., Liu, Y., and Wang, S. (2023a). A smart predict-then-optimize method for targeted and cost-effective maritime transportation. *Transportation Research Part B: Methodological*, 172:32–52. 3, 13, 19, 70, 129

Tian, X., Yan, R., Wang, S., and Laporte, G. (2023b). Prescriptive analytics for a maritime routing problem. *Ocean & Coastal Management*, 242:106695. 3, 13, 19, 70, 129

Tian, X., Yan, R., Wang, S., Liu, Y., and Zhen, L. (2023c). Tutorial on prescriptive analytics for logistics: What to predict and how to predict. *Electronic Research Archive*, 31(4):2265–2285. 3, 13, 19, 70, 129

Tian, Y., Wan, L., Han, K., and Ye, B. (2018). Optimization of terminal airspace operation with environmental considerations. *Transportation Research Part D: Transport and Environment*, 63:872–889. 134

Tran, T. T., Araujo, A., and Beck, J. C. (2016). Decomposition methods for the parallel machine scheduling problem with setups. *INFORMS Journal on Computing*, 28(1):83–95. 131

Tsang, M. Y., Shehadeh, K. S., and Curtis, F. E. (2023). An inexact column-and-constraint generation method to solve two-stage robust optimization problems. *Operations Research Letters*, 51(1):92–98. 73

Tsang, M. Y., Shehadeh, K. S., Curtis, F. E., Hochman, B. R., and Brentjens, T. E. (2024). Stochastic optimization approaches for an operating room and anesthesiologist scheduling problem. *Operations Research*. 73, 83, 92

Wang, H., Sun, Q., and Wang, S. (2024). Data-driven models for optimizing second-hand ship trading strategies under contextual information. *Naval Research Logistics*, 34:1–17. 13, 15, 20, 70

Wang, H., Yan, R., Au, M. H., Wang, S., and Jin, Y. J. (2023). Federated learning for green shipping optimization and management. *Advanced Engineering Informatics*, 56:101994. 13, 14, 20

Wang, K. and Jacquillat, A. (2020). A stochastic integer programming approach to air traffic scheduling and operations. *Operations Research*, 68(5):1375–1402. 130

Wang, R., Allignol, C., Barnier, N., Gondran, A., Gotteland, J.-B., and Mancel, C. (2022a). A new multi-commodity flow model to optimize the robustness of the gate allocation problem. *Transportation Research Part C: Emerging Technologies*, 136:103491. 1, 7, 8, 17, 18

Wang, S. and Yan, R. (2023). Fundamental challenge and solution methods in prescriptive analytics for freight transportation. *Transportation Research Part E: Logistics and Transportation Review*, 169:102966. 3, 13, 19, 70, 129

Wang, S., Zhao, C., Fan, L., and Bo, R. (2022b). Distributionally robust unit commitment with flexible generation resources considering renewable energy uncertainty. *IEEE Transactions on Power Systems*, 37(6):4179–4190. 72, 73, 92

Wang, X., Brownlee, A. E., Weiszer, M., Woodward, J. R., Mahfouf, M., and Chen, J. (2021). A chance-constrained programming model for airport ground movement optimisation with taxi time uncertainties. *Transportation Research Part C: Emerging Technologies*, 132:103382. 2

Wang, Z., Liang, M., and Delahaye, D. (2020). Automated data-driven prediction on aircraft estimated time of arrival. *Journal of Air Transport Management*, 88:101840. 33

Wu, H., Zhu, X., Li, S., Zhou, Y., Li, L., and Li, M. (2024). Distributionally robust ground delay programs with learning-driven airport capacity predictions. *arXiv preprint arXiv:2402.11415*. 21, 22, 39

Wu, L., Adulyasak, Y., Cordeau, J.-F., and Wang, S. (2022). Vessel service planning in seaports. *Operations Research*, 70(4):2032–2053. 23, 41, 73, 92

Xu, L., Zhang, C., Xiao, F., and Wang, F. (2017). A robust approach to airport gate assignment with a solution-dependent uncertainty budget. *Transportation Research Part B: Methodological*, 105:458–478. 2, 8, 11, 18, 19, 51

Xu, X., Gao, Y., Wang, H., Yan, Z., Shahidehpour, M., and Tan, Z. (2024). Distributionally robust optimization of photovoltaic power with lifted linear decision rule for distribution system voltage regulation. *IEEE Transactions on Sustainable Energy*, 15(2):758–772. 72

Yan, R., Wang, S., and Du, Y. (2020). Development of a two-stage ship fuel consumption prediction and reduction model for a dry bulk ship. *Transportation Research Part E: Logistics and Transportation Review*, 138:101930. 3, 21, 71

Yan, R., Wang, S., Zhen, L., and Jiang, S. (2024a). Classification and regression in prescriptive analytics: Development of hybrid models and an example of ship inspection by port state control. *Computers & Operations Research*, 163:106517. 3, 14, 20, 21, 71

Yan, R., Yang, D., Wang, T., Mo, H., and Wang, S. (2024b). Improving ship energy efficiency: Models, methods, and applications. *Applied Energy*, 368:123132. 13, 14, 20

Yan, R., Yang, Y., and Du, Y. (2022). Stochastic optimization model for ship inspection planning under uncertainty in maritime transportation. *Electronic Research Archive*, 31(4):103–122. 13, 14, 15, 16, 20, 21, 22, 36, 39, 70, 79, 142

Yang, Y., Yan, R., and Wang, S. (2024). Prescriptive analytics models for vessel inspection planning in maritime transportation. *Computers & Industrial Engineering*, 190:110012. 13, 14, 15, 16, 20, 21, 22, 36, 39, 70, 79, 129, 130, 142

Yin, Y., Wang, J., Chu, F., and Wang, D. (2024). Distributionally robust multi-period humanitarian relief network design integrating facility location, supply inventory and allocation, and evacuation planning. *International Journal of Production Research*, 62(1-2):45–70. 73

Yu, C., Zhang, D., and Lau, H. Y. (2017). An adaptive large neighborhood search heuristic for solving a robust gate assignment problem. *Expert Systems with Applications*, 84:143–154. 8, 18

Zhang, D. and Klabjan, D. (2017). Optimization for gate re-assignment. *Transportation Research Part B: Methodological*, 95:260–284. 47

Zhang, G., Jia, N., Zhu, N., He, L., and Adulyasak, Y. (2023). Humanitarian transportation network design via two-stage distributionally robust optimization. *Transportation Research Part B: Methodological*, 176:102805. 73

Zhang, Y., Zhang, Z., Lim, A., and Sim, M. (2021). Robust data-driven vehicle routing with time windows. *Operations Research*, 69(2):469–485. 72, 73