Transportation Research Part E 210 (2026) 104763

Contents lists available at ScienceDirect

Transportation Research Part E

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

Stabilised branch-and-check method for optimising multi-runway aircraft landing problem modelled through stochastic programming with learning-driven arrival time predictions

Chenliang Zhang, Zhongyi Jin, Kam K. H. Ng*, Ye Liu, Lingxiao Wu

Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong SAR, China

ARTICLE INFO

Keywords:
Multi-runway aircraft landing problem
Stochastic programming
Prescriptive analytics
Machine learning
Branch-and-check
Stabilisation.

ABSTRACT

The continuous growth in air traffic demand has strained the capacity of many hub airports. Effective runway operations are essential for optimising the utilisation of existing runways, and one of the key tasks is to manage runway operations under uncertainty. This paper examines the multi-runway aircraft landing problem (MALP) under arrival time uncertainty at hub airports with a multi-runway system, aiming to devise efficient and environmentally friendly aircraft landing operations. We model this problem using stochastic programming (SP) with a two-stage decision framework. With the support of advanced aviation technologies, air traffic control can typically make aircraft sequencing decisions based on precisely predicted arrival times. Therefore, we incorporate aircraft sequencing decisions into the second stage of the SP model. In the first stage, we assign aircraft to runways. In the second stage, we make aircraft sequencing and scheduling decisions after the arrival times are revealed. Given that the second-stage problem includes both integer and continuous variables, the model is termed the SP model with mixed-integer recourse (SP-MIR). We propose an optimisation-enhanced learning-driven scenario generation (OLSG) method, which employs machine learning techniques to estimate the distribution of unknown parameters, construct scenarios, and utilise the p-median problem to sample representative scenarios for input into the SP-MIR model. To efficiently solve the SP-MIR model, we present a novel exact decomposition approach, referred to as the stabilised branch-and-check (SBAC) method. This approach decomposes the original problem into a stabilised master problem and subproblems, wherein the master problem is stabilised around a designated stability centre to facilitate the generation of strong Benders cuts. The incorporated algorithmic enhancements further improve computational performance in solving both the stabilised master problem and subproblems. Our numerical experiment results demonstrate that the proposed SP-MIR model and OLSG method can enhance operational efficiency and reduce environmental impact in multi-runway operations for actual scenarios and real-world implementation. The results of scalability analysis indicate that the SBAC method and algorithmic enhancements achieve significant improvement in CPU time compared to well-known benchmark methods from the literature.

* Corresponding author.
E-mail addresses: poe-chenliang.zhang@connect.polyu.hk (C. Zhang), kimo-zhongyi.jin@connect.polyu.hk (Z. Jin), kam.kh.ng@polyu.edu.hk (K.K.H. Ng), ye.liu@connect.polyu.hk (Y. Liu), lingxiao-leo.wu@polyu.edu.hk (L. Wu).

https://doi.org/10.1016/j.tre.2026.104763
Received 6 February 2025; Received in revised form 11 December 2025; Accepted 10 February 2026
Available online 2 March 2026

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). A key challenge in reaching these improvements is solving the runway scheduling problem (RSP), which focuses on managing runway use to make sure aircraft take off and land safely and efficiently (Bennell et al., 2011; Ikli et al., 2021). The RSP includes the aircraft landing problem (ALP) for arriving aircraft, the aircraft take-off problem (ATP) for departing aircraft, and the aircraft sequencing and scheduling problem (ASSP), which considers both arrivals and departures. These problems have been extensively studied in deterministic settings, either assuming that aircraft arrival and departure times are known in advance or directly using estimated values (Harikiopoulo and Neogi, 2010; Bennell et al., 2011; Lieder et al., 2015; Sabar and Kendall, 2015; Salehipour, 2020; Ikli et al., 2021; Messaoud, 2021).

However, factors such as severe weather, air traffic delay propagation, technical challenges, and security considerations introduce variability in arrival and departure times, thereby introducing uncertainty into the input data for the RSP (Solak et al., 2018; Ng et al., 2017, 2020; Ikli et al., 2021; Khassiba et al., 2022). Indeed, a deterministic approach for the RSP that relies on estimated times, which may be easy to solve from a computational perspective, can produce sub-optimal runway scheduling decisions as it does not consider the uncertainty of the arrival and departure times. To address uncertainty in runway operations, modelling approaches such as robust optimisation (RO) and stochastic programming (SP) have been widely adopted, aiming to enhance the robustness and effectiveness of runway scheduling decisions (Sölveling et al., 2011a; Heidt et al., 2016; Kapolke et al., 2016; Ng et al., 2017; Solak et al., 2018; Khassiba et al., 2020, 2022).

Prior research on the RSP under uncertainty has mainly concentrated on single-runway and dual-runway systems in which arrivals and departures are handled separately (Sölveling et al., 2011a; Solak et al., 2018; Khassiba et al., 2020, 2022). However, due to increasing air traffic demand, many major international airports operate parallel multi-runway systems that adopt segregated operations, dedicating specific runways to arrivals or departures for enhanced safety and efficiency (Messaoud, 2021). This runway configuration requires air traffic control (ATC) to simultaneously determine aircraft-to-runway assignments, sequencing, and scheduling decisions, resulting in a more complex optimisation problem (Kapolke et al., 2016; Hong et al., 2018; Lieder et al., 2015; Salehipour, 2020; Sabar and Kendall, 2015; Messaoud, 2021). To this end, this paper investigates runway operations under uncertainty at airports with parallel multi-runway systems, with a particular focus on the multi-runway aircraft landing problem (MALP) under arrival time uncertainty. Our approach, however, can also be readily extended to the multi-runway aircraft take-off problem (MATP) with uncertain departure times. In this paper, we propose an SP model with a two-stage framework for the MALP under arrival time uncertainty, wherein aircraft-to-runway assignment is addressed in the first stage, and aircraft sequencing and landing time scheduling are jointly optimised in the second stage. Since the second-stage problem involves integer decision variables, the resulting formulation constitutes an SP with mixed-integer recourse (SP-MIR) model.

Existing SP approaches for the RSP have mainly modelled uncertainty using historical data or empirical knowledge (Sölveling et al., 2011a; Solak et al., 2018; Khassiba et al., 2020, 2022). However, such approaches often overlook contextual information, which can lead to suboptimal decisions (Ban and Rudin, 2019; Rahimian and Pagnoncelli, 2023; Sadana et al., 2025). More recent research proposes the use of machine learning (ML) techniques to incorporate contextual data when estimating the probability distributions of uncertain parameters, thereby enhancing the effectiveness of decision-making within SP approaches (Bertsimas and Kallus, 2020; Qi and Shen, 2022; Yan et al., 2022; Yang et al., 2024; Wang et al., 2025; Zhang et al., 2025b). This integration of predictive and optimisation approaches, commonly referred to as prescriptive analytics, enables data-driven decision-making informed by both historical and contextual information (Wang and Yan, 2023; Tian et al., 2023a,b,c; Sadana et al., 2025). In the aviation industry, ATC typically have access to extensive historical and contextual datasets, thereby facilitating the use of ML techniques to produce more accurate estimations of arrival and departure time distributions (Zhang et al., 2025a). Accordingly, this study adopts an estimate-then-optimise (ETO) approach, which is a subclass of prescriptive analytics, to infer probability distributions and construct scenarios to serve as inputs to the SP model. The proposed optimisation-enhanced learning-driven scenario generation (OLSG) method begins by estimating aircraft arrival times and constructing an initial scenario set. It then formulates a p-median problem to minimise the type-1 Wasserstein distance between the sampled scenarios and the original scenarios (Reese, 2006; Wang and Jacquillat, 2020). Representative scenarios are selected by the OLSG method through the p-median problem and then incorporated into the SP-MIR model for the MALP, thereby reducing computational complexity and facilitating more robust and efficient decision-making under uncertainty.

To efficiently solve the SP-MIR model for the MALP, we propose an exact decomposition technique termed the stabilised branch-and-check (SBAC) method, supported by algorithmic enhancements including lower bound lifting inequalities (LBLIs) and valid inequalities (VIs). The SBAC method decomposes the original problem into a stabilised master problem, which handles aircraft-to-runway assignment, and multiple subproblems addressing sequencing and scheduling decisions. In the stabilised master problem, trust region and reverse local branching constraints guide the search within a neighbourhood of a stability centre, facilitating the generation of strong Benders cuts and accelerating convergence. Operating over a single search tree, the SBAC method generates cuts at each feasible integer solution and applies them to unfathomed nodes, thereby avoiding redundant node evaluations.

The main contributions of this paper are summarised as follows:

(i) We develop the SP-MIR model for the MALP, aiming to devise efficient and environmentally friendly aircraft landing operations for multi-runway systems under uncertainty. This model minimises the weighted sum of expected makespan, fuel consumption and exhaust emission costs.

(ii) We introduce an OLSG method, which leverages ML techniques to estimate the distribution of uncertain parameters and construct an initial scenario set. This method then formulates a p-median problem to sample representative scenarios that preserve the underlying distributional structure, thereby enhancing the tractability and scalability of the SP-MIR model for the MALP.

(iii) We propose a novel exact decomposition method called the SBAC method for efficiently solving the SP-MIR model for the 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. The derived LBLIs and VIs further enhance computational efficiency in solving both the stabilised master problem and subproblems.

(iv) Extensive numerical experiments using real-world data from major international hub airports demonstrate the efficiency and environmental benefits of the proposed SP-MIR model for the MALP, supported by the OLSG method. Besides, scalability analyses show that the SBAC method and its algorithmic enhancements deliver significant improvements in computational performance, particularly in CPU time, compared to well-established benchmark methods from the literature.

The rest of this paper is structured as follows. Section 2 reviews the relevant research. Section 3 provides the problem setting and presents the formulations of the SP-MIR model for the MALP. Section 4 illustrates the scenario generation methods used in this study. Section 5 introduces the proposed SBAC method to solve the SP-MIR model for the MALP. In Section 6, numerical experiments are performed based on realistic data from major international hub airports. The results of the scalability analyses are reported in Section 7. Finally, the conclusions are presented in Section 8.

2. Literature review

2.1. RSP under uncertainty

The RSP aims to optimise aircraft operations on runways, subject to constraints such as wake vortex separation between successive aircraft and compliance with designated time windows. The RSP typically seeks to minimise total delay costs and makespan, as discussed in previous studies (Lieder et al., 2015; Bennell et al., 2017; Hong et al., 2017). Given the continued growth in air traffic and its environmental impact, especially the increase in emissions within terminal airspace that may affect public health in nearby regions, recent research has also considered fuel consumption and exhaust emission costs as additional optimisation objectives (Sölveling et al., 2011b; Bennell et al., 2017; Ikli et al., 2021; Chen et al., 2024).

Extensive research has addressed the RSP under deterministic aircraft arrival and departure times (Bennell et al., 2011; Solak et al., 2018; Ikli et al., 2021; Messaoud, 2021). Flight schedules are inherently uncertain due to factors such as severe weather, delay propagation, technical malfunctions, and security-related interventions (Solak et al., 2018; Ng et al., 2017, 2020; Ikli et al., 2021; Khassiba et al., 2022; Zhang et al., 2025a). Accordingly, uncertainty modelling approaches have been widely adopted to improve the robustness and reliability of decision-making in runway operations. Sölveling et al. (2011a) 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 the extended ALP on a single runway, where initial approach fixes are considered. Zhang et al. (2025a) employed prescriptive analytics for the RSP under uncertainty in single-runway operations, generating scenarios with an ML method and incorporating them into a distributionally robust optimisation (DRO) model to support decision-making and mitigate sampling errors.

Existing studies on the RSP under uncertainty have mainly focused on single-runway or dual-runway systems in which arrivals and departures are handled separately (Solak et al., 2018; Khassiba et al., 2020, 2022; Zhang et al., 2025a). In contrast, many hub international airports now operate with multiple parallel runways and adopt segregated parallel operations, whereby specific runways are designated for arrivals or departures to enhance efficiency and safety. Representative examples include Hartsfield-Jackson Atlanta International Airport, Hong Kong International Airport, Los Angeles International Airport, Shanghai Pudong International Airport, etc. This segregated parallel operation mode reduces inter-aircraft interaction, simplifies air traffic control, mitigates missed approaches, and lowers pilot error risks associated with incorrect instrument landing system selection. Runway operations in multi-runway systems entail not only sequencing and scheduling decisions but also runway assignment, thus significantly increasing computational complexity (Kapolke et al., 2016; Hong et al., 2018; Lieder et al., 2015; Salehipour, 2020; Sabar and Kendall, 2015; Messaoud, 2021). To investigate runway operations under uncertainty in multi-runway systems, this study adopts an SP approach to address the MALP, where arrival times are subject to uncertainty. The proposed framework is also readily extendable to the MATP under departure time uncertainty.

In operational settings, aircraft-to-runway assignments are made prior to terminal airspace entry, sequencing decisions are taken within the terminal area, and scheduling decisions are finalised during final approach. In existing SP and DRO models, arrival times are assumed to be known at final approach, with sequencing in the first stage and scheduling in the second (Solak et al.,

Table 1
Comparison of previous studies and our paper.

Paper	Runway configuration	Modelling approach	Decision variables	
			1st-stage	2nd-stage
Sölveling et al. (2011a)	Dual runways with segregated operations	SP-CR, reformulated as MILP	Sequencing	Scheduling
Solak et al. (2018)	Dual runways with segregated operations	SP-CR, reformulated as MILP	Sequencing	Scheduling
Khassiba et al. (2020)	Single runway dedicated to arrivals	SP-CR, reformulated as MILP	Sequencing and pre-tactical scheduling	Scheduling
Khassiba et al. (2022)	Single runway dedicated to arrivals	SP-CR, reformulated as MILP	Sequencing and pre-tactical scheduling	Scheduling
Zhang et al. (2025a)	Single runway with mixed operations	DRO-CR	Sequencing	Scheduling
Our paper	Multiple runways dedicated to arrivals	SP-MIR, reformulated as MILP	Runway assignment	Sequencing and scheduling

2018; Khassiba et al., 2020; Zhang et al., 2025a). The second-stage problems involve only continuous variables and are therefore referred to as continuous recourse, resulting in the SP-CR and DRO-CR formulations. However, advanced aviation technologies now support real-time monitoring and accurate arrival time prediction when aircraft are operating within the terminal airspace, thereby enabling sequencing decisions to be deferred to the second stage. Building on this technological advancement, we propose a revised SP model for the MALP under arrival time uncertainty, in which aircraft-to-runway assignment is addressed in the first stage and sequencing and scheduling are jointly optimised in the second. Since the second-stage problem involves integer decision variables, the resulting formulation is defined as an SP-MIR. Table 1 presents a comparative overview of previous studies employing the SP and DRO approaches for the RSP, alongside our proposed approach, with respect to runway configuration, modelling approach, and decision variables.

2.2. Prescriptive analytics

SP approaches often require the support of reasonably accurate estimates of the distributions of the uncertain parameters. Previous research on SP approaches for the RSP has predominantly relied on either unconditional probability distributions inferred from historical data (Sölveling et al., 2011a; Solak et al., 2018), or on empirical knowledge (Khassiba et al., 2020, 2022) to model uncertainties. However, prior approaches overlook contextual information, potentially resulting in suboptimal scheduling decisions and, in certain instances, compromising solution feasibility (Ban and Rudin, 2019; Rahimian and Pagnoncelli, 2023; Sadana et al., 2025). Contextual information, such as covariates or auxiliary features, can be effectively utilised by ML techniques to infer probability distributions of uncertain parameters. Driven by these precise estimations, subsequent optimisation techniques can generate plans that more effectively align with real-world operational conditions. 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).

The ETO approach is a promising paradigm within prescriptive analytics (Bertsimas and Kallus, 2020; Qi and Shen, 2022; Yan et al., 2022; Yang et al., 2024; Wang et al., 2025; Zhang et al., 2025b). Under this approach, the learning-driven scenario generation (LSG) method is employed to estimate the distribution of uncertain parameters, generate and sample scenarios. These sampled scenarios are subsequently utilised as inputs to SP models, enabling data-informed decision-making under uncertainty (Bertsimas and Kallus, 2020; Qi and Shen, 2022; Yan et al., 2022; Yang et al., 2024; Wang et al., 2025). Although the ETO approach holds promise for delivering satisfactory decisions, its effectiveness depends on the availability of a well-designed framework that needs to accurately incorporate the distributional characteristics of the estimated uncertain parameters into subsequent SP models. In the studies conducted by Yan et al. (2022) and Yang et al. (2024), k-nearest neighbours (KNN)-based ETO approaches were developed. These methods initially train the KNN algorithm to estimate the distribution of uncertain parameters, followed by the construction of a scenario set via the Cartesian product of the corresponding neighbourhood sets. As this Cartesian product typically yields an extensive number of scenarios, solving the entire set within an SP model becomes computationally intractable. To overcome this challenge, a subset of independent scenarios is randomly sampled from the original scenario set. The sample average approximation (SAA) method is then applied, assigning equal probability to each sampled scenario and approximating the expected objective by their average. In addition, Zhang et al. (2025b) proposed a random forest (RF)-based ETO approach and similarly applied the SAA method to construct a tractable deterministic approximation of the SP model using a finite set of sampled scenarios. The SAA method is commonly employed within the ETO framework to approximate the expected objective by averaging over a subset of independently sampled scenarios (Yan et al., 2022; Yang et al., 2024; Zhang et al., 2025b). It is important to note that the SAA method is asymptotically optimal, meaning its solution converges to the SP optimum as the number of scenarios approaches infinity (Kleywegt et al., 2002; Zhang et al., 2023). Nevertheless, the number of scenarios required to achieve SP optimum may become prohibitively large, thereby rendering the SP model computationally intractable. Furthermore, when applied to small sample sizes, the SAA method may produce optimistically biased estimates and lead to suboptimal solutions (Bertsimas and Kallus, 2020; Van Parys et al., 2021). To mitigate sampling errors, Zhang et al. (2025a) incorporated sampled scenarios into a DRO model, thereby improving decision quality for single-runway operations.

While DRO approaches can mitigate sampling errors by incorporating distributional ambiguity into the optimisation process, they often introduce tractability and scalability challenges in complex problems. To avoid these challenges, we select representative scenarios prior to optimisation to alleviate sampling errors, rather than handling them within the optimisation process as in Zhang et al. (2025a). By minimising the distance between the selected subset and the original scenario set, optimisation-based sampling methods provide a more suitable and computationally efficient alternative (Zhang et al., 2023). Hochreiter and Pflug (2007) formulated the scenario sampling task as a facility location problem, which minimises the Wasserstein metric to approximate stochastic processes. Wang and Jacquillat (2020) proposed a p-median problem for scenario sampling, which minimises the type-1 Wasserstein distance between the sampled and original scenarios, thereby enabling the identification of a more representative subset under limited sample size. Building upon these foundations, we introduce the OLSG method, which begins by estimating the distribution of aircraft arrival times to construct the scenario set. It then formulates a p-median problem to minimise the type-1 Wasserstein distance between the selected subset and the original scenario set. The type-1 Wasserstein distance imposes a linear penalty, which helps the OLSG method to preserve the representativeness of the sampled subset. These representative scenarios are subsequently used as inputs to the SP-MIR model for the MALP, thereby reducing computational burden and enhancing decision-making efficacy under uncertainty.

Related to our work, Zhang et al. (2025a) employed prescriptive analytics to optimise aircraft sequencing and scheduling for a single mixed-operation runway. We leverage one key idea from this study, which is the use of ML models to estimate uncertain parameters and generate scenarios via the Cartesian product of prediction outputs (see also Yan et al., 2022; Yang et al., 2024; Zhang et al., 2025b). However, our work differs in four main aspects: (i) multi-runway operations, as opposed to single-runway operations; (ii) MIR for the RSP under uncertainty, as opposed to CR; (iii) representative scenarios are selected prior to optimisation to enhance tractability while mitigating sampling and prediction errors, instead of being addressed within optimisation through DRO; and (iv) solution methods tailored to SP-MIR.

2.3. Decomposition method

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. Therefore, developing efficient solution methods for the SP-MIR model of the MALP is a pressing requirement.

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 Benders decomposition (BD) method is extensively used to solve SP models where the subproblems are required to be linear programming models with continuous variables (Khassiba et al., 2020; Rahmaniani et al., 2018). However, the studied SP-MIR model for the MALP considers the aircraft sequencing and scheduling problem in the second stage, which is a mixed-integer linear programming (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 these limitations, we propose the 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. While this stabilisation strategy has previously been employed within the BD method to solve the SP-CR model (Baena et al., 2020; Gong and Zhang, 2022), the present study is the first to adapt and implement it within the LBBD method to address the SP-MIR model. In addition, the initial integration of this strategy with the BAC method, where the stabilised master problem is formulated and solved only once within each neighbourhood, significantly reduces the CPU time required in solving the stabilised master problem. Furthermore, the use of LBLIs and VIs further enhances computational performance by respectively tightening the stabilised master problem and the subproblems in the SBAC method.

3. Stochastic programming model with mixed-integer recourse

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

Table 2
Notations.

Notation	Definition
Sets	
R	Set of runways.
I	Set of arriving aircraft.
Ω	Set of scenarios.
Parameters	
p^{ω}	Probability of scenario ω.
E_i^{ω}	The arrival time (i.e., earliest landing time) of aircraft i under scenario ω.
A^{ω}	The arrival time of the earliest aircraft under scenario ω.
S_{ij}	The separation time between aircraft $i \in I$ and $j \in I \setminus \{i\}$.
C_i^{Env}	Unit environmental cost of aircraft i.
W_1	The weight of the makespan.
W_2	The weight of the environmental costs.
M^{ω}	A sufficiently large number for the scenario ω.
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 directly precedes aircraft j on runway r under scenario ω; 0, otherwise.
t_i^{ω}	Landing time of aircraft i under scenario ω.
z^{ω}	Makespan under scenario ω.
d_i^{ω}	Delay time of aircraft i under scenario ω.

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 the 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^{ω}. The arrival time of aircraft $i \in I$, denoted by E_i^{ω}, represents its earliest landing time under scenario $\omega \in \Omega$. The landing time t_i^{ω} of aircraft i must not be earlier than its arrival time E_i^{ω}, and the delay time d_i^{ω} is the difference between these two values. Each runway can process, at most, one aircraft at a time. We formulate the MALP as a directed network $G = (V, A)$, which enables the explicit representation of aircraft sequencing decisions on each runway. The vertex set $V = I \cup \{s, e\}$, where s and e are dummy starting and ending aircraft, respectively. An arc $(i, j) \in A$ exists if aircraft $i \in I \cup \{s\}$ can be scheduled to land immediately before aircraft $j \in I \cup \{e\} \setminus \{i\}$ on the same runway. The binary variable $y_{ij}^{r\omega}$ is equal to 1 if aircraft i immediately precedes aircraft j on the same runway r under scenario ω. The inclusion of dummy aircraft clearly identifies the start and end of each runway's aircraft landing sequence, enabling the definition of a complete and well-structured sequence. If aircraft i and j are assigned to the same runway sequentially, a separation time S_{ij} is required between the landing times of aircraft to ensure the wake vortex separation requirements. The arrival time of the earliest arriving aircraft under scenario ω is denoted as A^{ω}, where $A^{\omega} = \min_{i \in I} \left\{ E_i^{\omega} \right\}$. The variable z^{ω} represents the makespan under scenario ω, defined as the difference between the landing time of the last aircraft to use the runway system and A^{ω}.

The SP-MIR model for the MALP aims to minimise the weighted sum of the expected makespan and environmental costs. In this paper, weights W_1 and W_2 are respectively used to reflect the preferences for makespan and environmental costs, where $W_1 + W_2 = 1$. Environmental costs primarily encompass fuel consumption costs and exhaust emission costs. The cost of fuel consumption is determined by the unit price of the jet fuel C^{Fuel}, the fuel burn rate α_i, and the delay time d_i^{ω} (Sölveling et al., 2011b). Thus, the additional fuel cost incurred by aircraft i under scenario ω is expressed as $C^{\mathrm{Fuel}} \alpha_i d_i^{\omega}$. Exhaust emissions mainly consist of CO_2, CO, HC, NO_x, and SO_2 (Sölveling et al., 2011b; Tian et al., 2018). Following the exhaust emission modelling method proposed by Sölveling et al. (2011b), the emissions of CO_2 are assumed to be proportional to the fuel flow $\alpha_i d_i^{\omega}$ with a factor β. For the remaining pollutants $m \in \{\mathrm{CO}, \mathrm{HC}, \mathrm{NO}_x, \mathrm{SO}_2\}$, emissions are estimated by multiplying the respective emission rates ε_i^m by the delay time d_i^{ω}. Each emission type is associated with an external cost, denoted by C^m for pollutant m. Accordingly, the unit environmental cost attributable to aircraft i is formulated as: $C_i^{\mathrm{Env}} = C^{\mathrm{Fuel}} \alpha_i + C^{\mathrm{CO}_2} \beta \alpha_i + \sum_{m \in \{\mathrm{CO}, \mathrm{HC}, \mathrm{NO}_x, \mathrm{SO}_2\}} C^m \varepsilon_i^m$. We present the notations for sets, parameters, and decision variables in Table 2.

Given the problem description and notations, the SP-MIR model for the MALP is provided as follows:

$$\min \quad \sum_{\omega \in \Omega} p^{\omega} \left(W_1 z^{\omega} + W_2 \sum_{i \in I} C_i^{\mathrm{Env}} d_i^{\omega} \right) \tag{1a}$$

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

$$x_i^r = \sum_{j \in I \cup \{e\} \setminus \{i\}} y_{ij}^{r\omega}, \quad \forall r \in R, \forall i \in I, \forall \omega \in \Omega, \tag{1c}$$

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

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

$$\sum_{j \in I \cup \{s\} \setminus \{i\}} y_{ji}^{r\omega} = \sum_{j \in I \cup \{e\} \setminus \{i\}} y_{ij}^{r\omega}, \quad \forall r \in R, \forall i \in I, \forall \omega \in \Omega, \tag{1f}$$

$$t_i^\omega \geq E_i^\omega, \quad \forall i \in I, \forall \omega \in \Omega, \tag{1g}$$

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

$$z^\omega \geq t_i^\omega - A^\omega, \quad \forall i \in I, \forall \omega \in \Omega, \tag{1i}$$

$$d_i^\omega \geq t_i^\omega - E_i^\omega, \quad \forall i \in I, \forall \omega \in \Omega, \tag{1j}$$

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

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

$$t_i^\omega \in \mathbb{R}^+, \quad \forall i \in I, \forall \omega \in \Omega, \tag{1m}$$

$$z^\omega \in \mathbb{R}^+, \quad \forall \omega \in \Omega, \tag{1n}$$

$$d_i^\omega \in \mathbb{R}^+, \quad \forall i \in I, \forall \omega \in \Omega. \tag{1o}$$

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

4. Scenario generation methods

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., 2011b; Solak et al., 2018). Next, we propose the LSG method based on the RF method. In this approach, after the RF method determines the distribution of aircraft arrival times, the Cartesian product is used to construct all possible value combinations to form a scenario set. Since the original scenario set may be very large and make the SP model intractable, we adopt the SAA method, in which a scenario subset is randomly sampled with equal probability from the original set to construct a tractable approximation of the SP model. Finally, based on the LSG method, we further develop the OLSG method. To address the limitations of the SAA technique used in the LSG method, including scalability challenges with large scenario sets and reduced reliability under small-sample conditions where estimation bias may occur, the OLSG method introduces a p-median problem to select a representative subset of scenarios.

4.1. 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., 2011b; Solak et al., 2018). In this study, the HDSG method first analyses

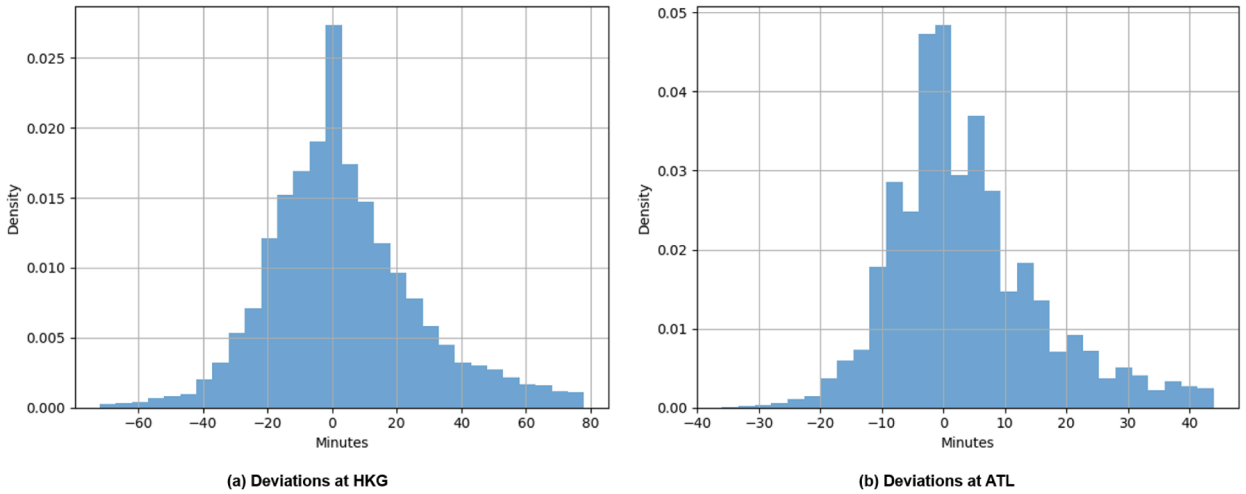


Fig. 1. Histograms of arrival time deviations at HKG and ATL.

the deviation between the estimated and actual aircraft arrival times using historical data. Subsequently, a scenario set Ω is generated based on the analysed distribution, with each scenario independently sampled and assigned equal probability. This sampled scenario set is then utilised in the subsequent SP-MIR model for the MALP.

Using data from 1 to 30 October 2023 at Hong Kong International Airport (IATA: HKG) and Hartsfield-Jackson Atlanta International Airport (IATA: ATL), we present the histograms of deviations between estimated and actual arrival times, as shown in Fig. 1. Utilising data from 1st to 30th October 2023, we present the distribution of deviations between estimated and actual arrival times in Fig. 1. 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 Fig. 1. All sampled scenarios are included in the scenario set Ω. The probability of each scenario in set Ω is $\frac{1}{|\Omega|}$.

4.2. Learning-driven scenario generation method

In the LSG method, 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 analytics research (Bertsimas and Kallus, 2020; Wang et al., 2025; Zhang et al., 2025b). 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 well it predicts aircraft arrival times. The training process details are provided in Appendix A.

We then estimate the distributions of arrival times through the trained RF method and generate scenarios for the SP-MIR model of the MALP based on these estimations. 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 $\mathscr{P}(f_i)$. The arrival time vector for set I is written as $\widetilde{\boldsymbol{E}} = \left(\widetilde{E}_1, \widetilde{E}_2, \dots, \widetilde{E}_{|I|}\right)$. Following previous works (Yang et al., 2024; Zhang et al., 2025b), we employ the Cartesian product $\Phi(\widetilde{\boldsymbol{E}}) = \left\{\mathscr{P}(f_1) \times \mathscr{P}(f_2) \times \cdots \times \mathscr{P}(f|I|)\right\}$ to approximate the distribution of $\widetilde{\boldsymbol{E}}$. This product includes $N^{|I|}$ elements, which grows exponentially with $|\boldsymbol{I}|$ (Yan et al., 2022; Zhang et al., 2025b). Accordingly, the uncertain arrival times are represented by a set of scenarios, $\Xi = \{1, 2, \dots, |\Phi(\widetilde{E})|\}$, with size $N^{|I|}$.

Considering all scenarios in Ξ would make solving the subsequent SP-MIR model overly complex. Thus, following the approach of Yang et al. (2024), we approximate the distribution of aircraft arrival times by randomly sampling a subset of scenarios, Ω, from Ξ, with each scenario having an equal probability $\frac{1}{|\Omega|}$.

4.3. Optimisation-enhanced learning-driven scenario generation method

In this section, we introduce the OLSG method for selecting a representative subset of scenarios. The overall framework is depicted in Fig. 2. The key difference is that the OLSG method employs the p-median problem for scenario sampling, aiming to minimise the type-1 Wasserstein distance between the selected subset and the original scenario set, thereby ensuring that the sampled scenarios are representative. In the OLSG method, we first use the RF method to generate the scenario set Ξ. The p-median problem is then employed to select $|\Omega|$ scenarios from Ξ. However, as noted in Subsection 4.2, Ξ 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 randomly sample a subset of scenarios Ξ^{sub} from the original scenario set Ξ. Following this, the p-median problem (2) is employed to sample the scenario set Ω from Ξ^{sub}, which is subsequently used in the SP model. In the OLSG method, the set size $|\Omega|$ is defined

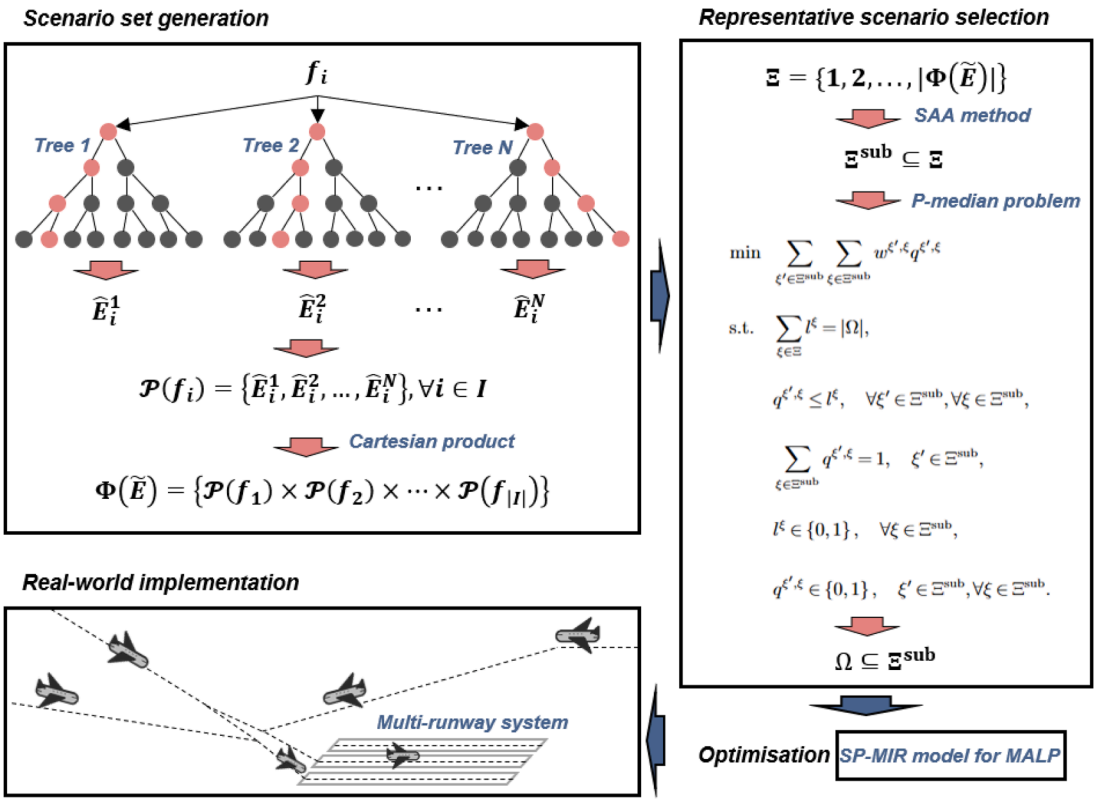


Fig. 2. Framework of the OLSG method.

in advance, and each scenario in the set Ω has an equal probability $\frac{1}{|\Omega|}$. The size of the subset Ξ^{sub} extracted from the original set Ξ is defined as $|\Xi^{\text{sub}}| = \gamma|\Omega|$, where γ 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 Ω are randomly selected from Ξ, with each scenario in Ω assigned an equal probability of $\frac{1}{|\Omega|}$. As the value of γ increases, the scenarios in Ω need to be sampled from a larger set Ξ^{sub} through the p-median problem.

In the p-median problem, we minimise the type-1 Wasserstein distance between the sampled subset Ω and the original scenario set Ξ^{sub}. The parameter $w^{\xi',\xi}$ denotes the type-1 Wasserstein distance between scenario ξ' and ξ. The number of scenarios selected into the set Ω is constrained by the parameter $|\Omega|$. A binary decision variable l^{ξ} equals 1 if scenario $\xi' \in \Xi^{\text{sub}}$ is incorporated into the scenario set Ω, 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:

$$\min \quad \sum_{\xi' \in \Xi^{\text{sub}}} \sum_{\xi \in \Xi^{\text{sub}}} w^{\xi',\xi} q^{\xi',\xi} \tag{2a}$$

$$\text{s.t.} \quad \sum_{\xi \in \Xi^{\text{sub}}} l^{\xi} = |\Omega|, \tag{2b}$$

$$q^{\xi',\xi} \le l^{\xi}, \quad \forall \xi' \in \Xi^{\text{sub}}, \forall \xi \in \Xi^{\text{sub}}, \tag{2c}$$

$$\sum_{\xi \in \Xi^{\text{sub}}} q^{\xi',\xi} = 1, \quad \xi' \in \Xi^{\text{sub}}, \tag{2d}$$

$$l^{\xi} \in \{0,1\}, \quad \forall \xi \in \Xi^{\text{sub}}, \tag{2e}$$

$$q^{\xi',\xi} \in \{0,1\}, \quad \xi' \in \Xi^{\text{sub}}, \forall \xi \in \Xi^{\text{sub}}. \tag{2f}$$

The objective function (2a) minimises the type-1 Wasserstein distance. Constraints (2b) ensure $|\Omega|$ scenarios are chosen from Ξ^{sub}. Constraints (2c) guarantee that all scenarios $\xi' \in \Xi^{\text{sub}}$ are mapped to one of the selected scenarios in Ω. Constraints (2d) make sure

that each scenario ξ' is assigned to a scenario in Ω. Constraints (2e) and (2f) define the range of the decision variables. The scenarios selected in the set Ω have equal probability $\frac{1}{|\Omega|}$.

5. SBAC method with algorithmic enhancements

5.1. SBAC method

Because of the NP-hard nature of the MALP, existing commercial mixed-integer programming (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 the 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 that the Hamming distance between any candidate integer solution and the stability centre $\hat{x}^{sc}$ remains within a predefined trust region radius κ, where $\kappa \geq 1$. The Hamming distance is defined as the number of positions at which two binary strings of equal length differ. Specifically, they restrict the number of binary variables permitted to switch values to at most κ components. The radius κ can be dynamically adjusted throughout the solution process. Besides, the reverse local branching constraints avoid the SBAC method repeatedly searching the neighbourhood κ 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 promptly obtain an effective global lower bound for the original problem.

The implementation framework of the SBAC method is presented in Algorithm 1. We set the lower bound (LB), upper bound (UB), neighbourhood capacity nc, and trust region radius κ during the initialisation phase. Additionally, we define the set $\mathcal{F}$, cut pool $\mathcal{M}$, and $\mathcal{N}$ as empty sets. By solving Model (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 (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 that an integer feasible solution $\hat{x}^{\zeta}$ is found. The disaggregated subproblems (3) are then solved with this solution. Subsequently, the resulting Benders cuts (4) and (5) are added to the unfathomed nodes. Additionally, the Benders cuts (4) and (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 (6) is found. The UB is updated with the related optimal solution value, and the stabilised master problem (6) without the trust region constraints (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 (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 that the stabilised master problem (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 (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 κ. The initial value of κ can be set to a number greater than 0 and less than or equal to nc. When the stabilised master problem (6) becomes infeasible, the value of κ is increased to a value greater than the current value of κ and less than or equal to nc. In Subsection 7.1, 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 1 can solve the optimisation problem within a finite number of iterations:

Proposition 1. *The SBAC method can solve the SP-MIR model for the 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 will be added to the stabilised master problem during the search. The values of trust region radius κ 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. □

5.1.1. 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 (6) is solved at iteration ζ, 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 ζ, 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

Algorithm 1: 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 (6) with $\hat{x}^{sc}$, $\mathcal{F}$, $\mathcal{M}$ and $\mathcal{N}$ using a MIP solver.
4: **If** the stabilised master problem (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 κ, go to line 3.
9: **Else**
10: **If** an optimal solution $\hat{x}^{so}$ of stabilised master problem (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 (3) without trust region constraints (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 (3) with $\hat{x}^{\zeta}$.
20: Add Benders cuts (4) and (5) to the unfathomed nodes.
21: Add Benders cuts (4) and (5) to cut pools $\mathcal{M}$ and $\mathcal{N}$, respectively.
22: **End if**
23: **End if**

r under scenario ω can be formulated as model (3), where the r superscript of y is ignored. Moreover, as the aircraft cluster $\hat{\mathcal{H}}^{r\zeta}$ assigned to runway r is already known, the value of M^{ω} used in Constraints (1h) can be tightened accordingly. Specifically, within Constraints (3g), M^{ω} can be refined to $M^{r\omega\zeta} = \max_{i\in\hat{\mathcal{H}}^{r\zeta}} \left\{E_i^{\omega}\right\} + \max_{i\in\hat{\mathcal{H}}^{r\zeta}, j\in\hat{\mathcal{H}}^{r\zeta}, i\neq j} \left\{S_{ij}\right\}\left(|\hat{\mathcal{H}}^{r\zeta}| - 1\right)$, which provides a tighter bound for the Big-M constant.

$$\min \quad W_1 z^{\omega} + W_2 \sum_{i\in\hat{\mathcal{H}}^{r\zeta}} C_i^{\mathrm{Env}} d_i^{\omega} \tag{3a}$$

$$\text{s.t.} \quad \sum_{j\in\hat{\mathcal{H}}^{r\zeta}\cup\{e\}\setminus\{i\}} y_{ij}^{\omega} = 1, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{3b}$$

$$\sum_{i\in\hat{\mathcal{H}}^{r\zeta}} y_{si}^{\omega} = 1, \tag{3c}$$

$$\sum_{i\in\hat{\mathcal{H}}^{r\zeta}} y_{ie}^{\omega} = 1, \tag{3d}$$

$$\sum_{j\in\hat{\mathcal{H}}^{r\zeta}\cup\{s\}\setminus\{i\}} y_{ji}^{\omega} = \sum_{j\in\hat{\mathcal{H}}^{r\zeta}\cup\{e\}\setminus\{i\}} y_{ij}^{\omega}, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{3e}$$

$$t_i^{\omega} \geq E_i^{\omega}, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{3f}$$

$$t_i^{\omega} + S_{ij} - t_j^{\omega} \leq M^{r\omega\zeta}(1 - y_{ij}^{\omega}), \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \forall j \in \hat{\mathcal{H}}^{r\zeta}, i \neq j, \tag{3g}$$

$$z^{\omega} \geq t_i^{\omega} - A^{\omega}, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{3h}$$

$$d_i^{\omega} \geq t_i^{\omega} - E_i^{\omega}, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{3i}$$

$$y_{ij}^{\omega} \in \{0,1\}, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \forall j \in \hat{\mathcal{H}}^{r\zeta}, i \neq j, \tag{3j}$$

$$t_i^{\omega} \in \mathbb{R}^+, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}, \tag{3k}$$

$$z^{\omega} \in \mathbb{R}^+, \tag{3l}$$

$$d_i^{\omega} \in \mathbb{R}^+, \quad \forall i \in \hat{\mathcal{H}}^{r\zeta}. \tag{3m}$$

5.1.2. 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 the 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 ω at iteration ζ. Then, the related Benders optimality cuts are written as Eq. (4) and (5). The value of M is set to $3\hat{\mathcal{Z}}^{r\omega\zeta}$ for Eq. (4) and $3\hat{\mathcal{C}}^{r\omega\zeta}$ for Eq. (5), respectively. In subsequent iterations, cuts (4) ensure that if all aircraft in the set $\hat{\mathcal{H}}^{r\zeta}$ are assigned to runway r, the makespan under scenario ω should be greater than or equal to $\hat{\mathcal{Z}}^{r\omega\zeta}$. Similarly, cuts (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 ω should be no less than $\hat{\mathcal{C}}^{r\omega\zeta}$.

$$\eta^{\omega} \geq \hat{\mathcal{Z}}^{r\omega\zeta} - M \sum_{i \in \hat{\mathcal{H}}^{r\zeta}} (1 - x_i^r), \quad \forall r \in R, \forall \omega \in \Omega, \tag{4}$$

$$\theta^{r\omega} \geq \hat{\mathcal{C}}^{r\omega\zeta} - M \sum_{i \in \hat{\mathcal{H}}^{r\zeta}} (1 - x_i^r), \quad \forall r \in R, \forall \omega \in \Omega. \tag{5}$$

5.1.3. Stabilised master problem

The initial stabilised master problem is written as follows:

$$\min \quad \sum_{\omega \in \Omega} p^{\omega} \left(W_1 \eta^{\omega} + W_2 \sum_{r \in R} \theta^{r\omega} \right) \tag{6a}$$

$$\text{s.t.} \quad (1b), \ (1k), \tag{6b}$$

$$\sum_{(r,i): \hat{x}_i^{r,sc}=1} (1 - x_i^r) + \sum_{(r,i): \hat{x}_i^{r,sc}=0} x_i^r \leq \kappa, \tag{6c}$$

$$\sum_{(r,i): \hat{x}_i^{r,sc'}=1} (1 - x_i^r) + \sum_{(r,i): \hat{x}_i^{r,sc'}=0} x_i^r \geq \kappa' + 1, \quad \forall (\hat{x}^{sc'}, \kappa') \in \mathcal{F}, \tag{6d}$$

$$\eta^{\omega} \geq \hat{\mathcal{Z}}^{m\omega} - M \sum_{i \in \hat{\mathcal{H}}^m} (1 - x_i^r), \quad \forall m \in \mathcal{M}, \forall r \in R, \forall \omega \in \Omega, \tag{6e}$$

$$\theta^{r\omega} \geq \hat{\mathcal{C}}^{n\omega} - M \sum_{i \in \hat{\mathcal{H}}^n} (1 - x_i^r), \quad \forall n \in \mathcal{N}, \forall r \in R, \forall \omega \in \Omega, \tag{6f}$$

$$\eta^{\omega} \in \mathbb{R}^+, \quad \forall \omega \in \Omega, \tag{6g}$$

$$\theta^{r\omega} \in \mathbb{R}^+, \quad \forall r \in R, \forall \omega \in \Omega. \tag{6h}$$

The trust region constraints (6c) guarantee that the integer solutions identified in the branch-and-cut tree remain within a trust region radius κ of a stability centre $\hat{x}^{sc}$. The reverse local branching constraints (6d) are used to avoid the SBAC method repeatedly searching the neighbourhood κ' 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 κ'. Constraints (6e) and (6f) are initial cuts for adding the previously found Benders cuts to the root node of the new branch-and-cut tree. Constraints (6g) and (6h) define the domain of decision variables.

5.2. Algorithmic enhancements

To enhance the computational efficiency of the proposed SBAC method, this subsection introduces two algorithmic enhancements known as LBLIs and VIs.

5.2.1. LBLIs

In the SBAC method, since parts of the original objective function are projected out in the master problem, the initial LB may be weak, resulting in a large optimality gap during the early stages and requiring a substantial number of Benders cuts to close the gap (Adulyasak et al., 2015; Rahmaniani et al., 2017, 2018; Wu et al., 2022). In this study, the LBLIs incorporate information from the original objective function and are introduced as initial cuts in the stabilised master problem to improve the LB, specifically by representing the LB of runway assignment costs.

To generate the LBLIs, we first systematically construct all unique unordered pairs from the set I. Specifically, we generate each pair $\{i,j\}$ with $i \neq j$, ensuring that neither $\{i,j\}$ nor its symmetric counterpart $\{j,i\}$ has been previously included. This procedure guarantees a complete set of distinct combinations, excluding duplicates and self-pairings. We define $L = \{1,2,\dots,|L|\}$ as the index set of $|L|$ distinct pairs, each represented by an unordered pair $\{i,j\}$. Subsequently, we set $\hat{\mathcal{H}}^{\ell} = \{i,j\}$ and solve the corresponding disaggregated subproblem under scenario ω. This yields the associated makespan $\hat{\mathcal{Z}}^{\ell\omega}$ and total environmental cost $\hat{C}^{\ell\omega}$. The resulting LBLIs are formulated as follows:

$$\eta^{\omega} \geq \hat{\mathcal{Z}}^{\ell\omega} - M \sum_{i \in \hat{\mathcal{H}}^{\ell}} (1 - x_i^r), \quad \forall \ell \in \mathcal{L}, \forall r \in R, \forall \omega \in \Omega. \tag{7}$$

$$\theta^{r\omega} \geq \hat{C}^{\ell\omega} - M \sum_{i \in \hat{\mathcal{H}}^{\ell}} (1 - x_i^r), \quad \forall \ell \in \mathcal{L}, \forall r \in R, \forall \omega \in \Omega. \tag{8}$$

5.2.2. VIs

The disaggregated subproblems in the SBAC method are formulated as MILP, which must be solved in large quantities and are typically computationally demanding. To alleviate this burden, we introduce a set of VIs designed to estimate the LB of arc costs within the aircraft landing sequence. These inequalities aim to provide a tighter LB estimate and thereby accelerate the solution process of each disaggregated subproblem. For aircraft $i \in I$ and aircraft $j \in I \setminus \{i\}$ that are scheduled to land consecutively on the same runway, the LB of delay time incurred by arc (i,j) under scenario ω is given by $\mathcal{D}_{ij}^{\omega} = \max\left\{0, \left(E_i^{\omega} + S_{ij} - E_j^{\omega}\right)\right\}$. The VIs in (9) can be incorporated into the disaggregated subproblem (3) for each scenario $\omega \in \Omega$ to strengthen the LB on delay time.

$$d_j^{\omega} \geq \mathcal{D}_{ij}^{\omega} y_{ij}^{\omega}, \quad \forall i \in I, \forall j \in I, i \neq j. \tag{9}$$

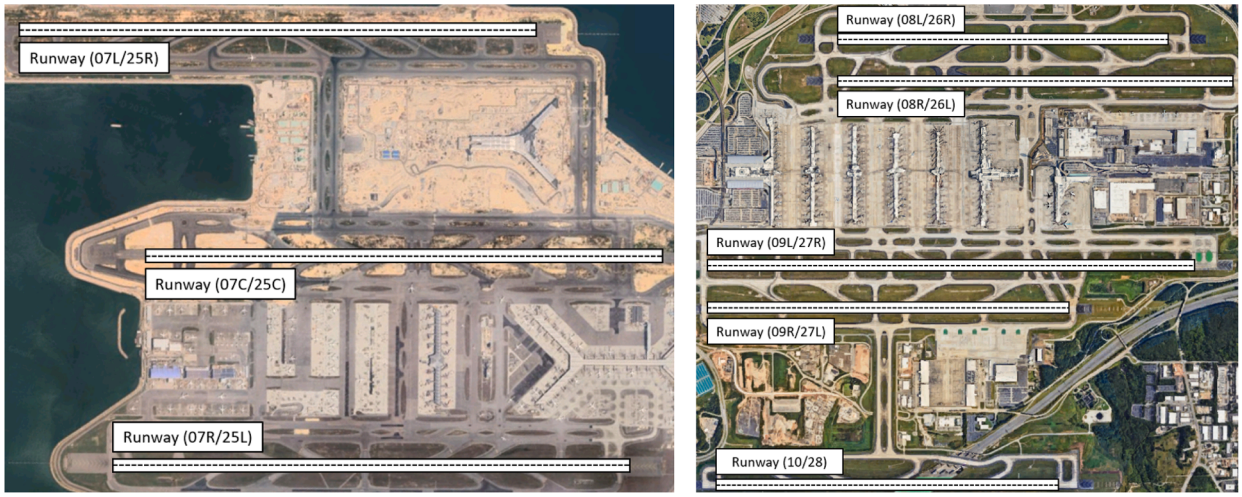
6. Numerical experiments

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 the MALP, supported by the OLSG method. We then compare the performance of the SP-MIR and SP-CR models for the MALP. Finally, we examine the impact of the decision-maker preference levels.

6.1. Experimental design

The experiments are conducted using real-world data from HKG and ATL, two major international hub airports. HKG operates a three-runway system, as illustrated in Fig. 3(a), where the northern runway (07L/25R) is designated for landings, the central runway (07C/25C) is assigned for takeoffs, and the southern runway (07R/25L) operates in a flexible mode, accommodating either takeoffs or landings depending on hourly traffic demand. ATL features a five-runway configuration, as shown in Fig. 3(b), with runways 8L/26R and 9L/27R primarily used for landings, runways 8R/26L and 9R/27L designated for takeoffs, and runway 10/28 functioning as a flexible runway that adjusts its mode based on real-time traffic conditions.

The test instances are generated based on real operational data from HKG and ATL collected on 30 October 2023. Detailed information for each instance is provided in Appendix B. We sample 10, 50, and 100 independent scenarios for each instance using the scenario generation methods described in Section 4. We set an upper limit of 600 seconds for the CPU time of the OLSG methods. Fig. 4 presents the average CPU time required by the OLSG methods to generate scenarios for the 135 test instances. The LSG method is considered a special case of the OLSG method with a selection ratio $\gamma = 1$ in this figure. In general, we observe that the OLSG method requires more CPU time as the number of $|\Omega|$ increases. Additionally, as the value of γ increases, i.e., when scenarios in Ω are selected from larger Ξ^{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), the CPU times required to generate scenario sets Ω 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 Ω of 10, 50, and 100, the required CPU times are 0.07, 3.64, and 26.08 seconds, respectively, still within an acceptable range. Lastly, when $\gamma = 10$, the required CPU time remains small at 0.66 seconds for a scenario set size Ω of 10. However, when Ω is 50 or 100, the required CPU times increase significantly to 345.06 and 593.43 seconds, respectively. In addition to computational performance, Appendix C provides an out-of-sample analysis to assess how well OLSG preserves the statistical characteristics of the uncertain parameters.



(a) Three-runway system at HKG. (b) Five-runway system at ATL.

Fig. 3. Runway systems at HKG and ATL.

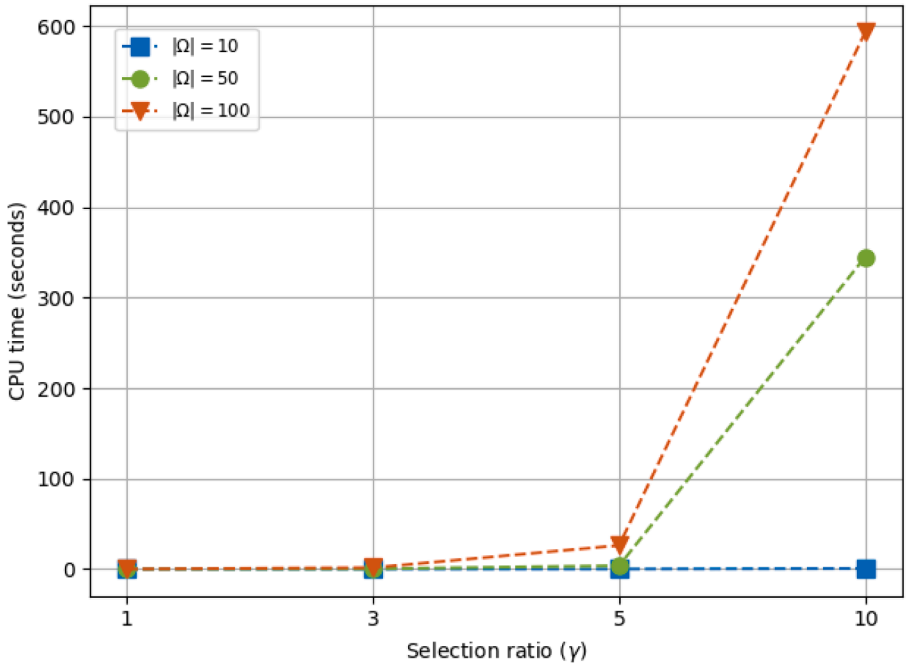


Fig. 4. CPU time required by the OLSG method.

Serhan et al. (2018) provided the fuel burn rate for each aircraft type. As this paper 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. (2011b), we provide the emission rates and costs. The emission amount of CO_2 is proportional to the fuel flow, and the factor β is 3.14, i.e., 1 lb of jet fuel emits 3.14 lb of CO_2. Table 3 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 4 presents the separation time requirements for all aircraft type combinations, following the setting of Pohl et al. (2021). Given that different selection ratios γ are considered in the OLSG method, we use the notation $OLSG_\gamma$ to represent the OLSG method under various γ values.

Table 3
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

Table 4
Separation time requirements (in seconds).

Leading	Trailing	
	Heavy	Large
Heavy	96	157
Large	60	69

6.2. Actual sample analysis

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 ω. In the actual sample analysis, we solve the second-stage problem using the optimal solution $\hat{\mathbf{x}}$ under the actual scenario ω^{actual}, thereby obtaining $Q(\hat{\mathbf{x}}, \omega^{\text{actual}})$. Here, ω^{actual} represents the realised operational conditions, including the actual arrival times of all aircraft in the test instances. 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, $OLSG_3$, $OLSG_5$, and $OLSG_{10}$ methods. The average results obtained from 135 test instances, as presented in Fig. 5, illustrate the performance of each optimisation approach within 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 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. Moreover, when the number of scenarios increases to 100, the performance deteriorates compared to the case with 50 scenarios. This may be because the distribution of uncertain parameters cannot be accurately estimated using historical data alone. As the number of scenarios increases, the scenario generation process may produce several extreme or unrealistic scenarios due to the misestimated distribution. These scenarios are incorporated into the SP-MIR model, potentially resulting in suboptimal decisions under actual scenarios. Moreover, all SP-MIR models supported by different scenario generation methods consistently outperform the deterministic model.

Table 5 presents the detailed results of the actual sample analysis. The deterministic benchmark model yields a makespan of 648.94 and environmental costs of 238.23. Across all tested methods and scenario sizes, the makespan remains relatively stable, fluctuating between approximately 630 and 650. In contrast, environmental costs exhibit more substantial variation and follow a similar trend to the previously discussed total objective value. With 10 scenarios, the HDSG method achieves an 18.52% reduction in environmental costs compared to the deterministic model. In contrast, other methods incur environmental costs approximately 21.35% to 26.26% higher than the benchmark. When the number of scenarios increases to 50, the SP-MIR model, incorporating various scenario generation techniques, outperforms the deterministic model, with reductions in environmental costs ranging from 25.75% to 31.03%. A further increase to 100 scenarios results in even greater improvements, with environmental cost reductions ranging from 26.39% to 41.20%. Moreover, all scenario generation methods supported by the ML-based framework demonstrate a consistent downward trend in environmental costs as the number of scenarios increases. Specifically, when increasing the scenario count from 10 to 100, the LSG method achieves a reduction of 6.60%, $OLSG_3$ achieves 9.37%, $OLSG_5$ achieves 20.25%, and $OLSG_{10}$ achieves the largest reduction of 24.80%.

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 0.24% higher than the best. It is worth noting that generating 100 scenarios using the $OLSG_{10}$ method takes 593.43 seconds, which is 95.61% longer than the 26.08 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.

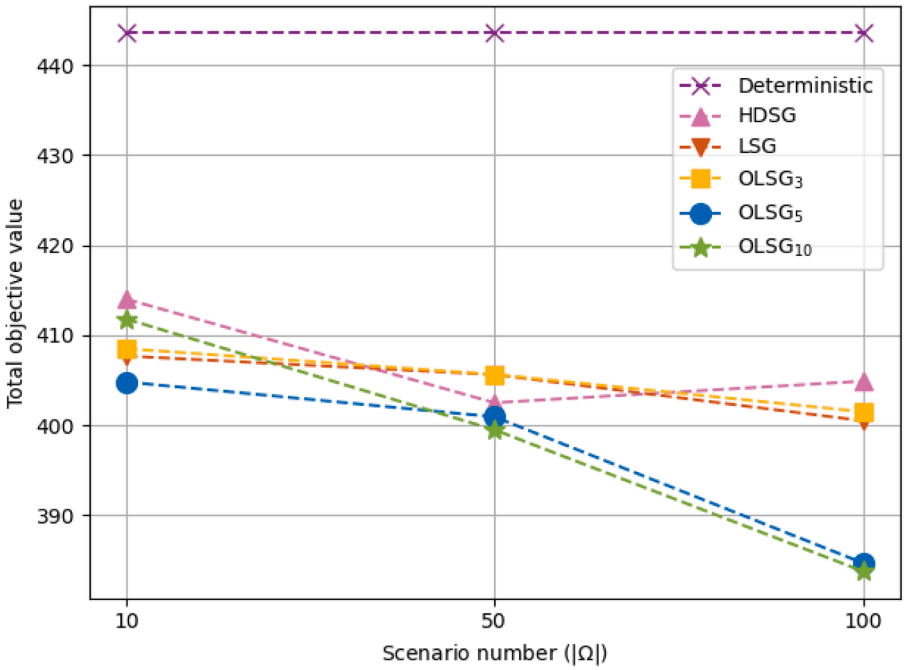


Fig. 5. Overall results of the actual sample analysis.

Table 5
Detail results of the actual sample analysis.

	$\|\Omega\| = 10$		$\|\Omega\| = 50$		$\|\Omega\| = 100$	
	Makespan	Environmental	Makespan	Environmental	Makespan	Environmental
HDSG	633.91	194.10	634.47	170.52	634.44	175.35
LSG	636.97	178.41	634.36	176.90	634.36	166.63
$OLSG_3$	632.96	184.06	636.31	174.98	636.18	166.81
$OLSG_5$	633.94	175.66	633.89	167.97	629.30	140.09
$OLSG_{10}$	636.24	187.36	634.66	164.32	626.66	140.90

6.3. Real-world implementation

In this subsection, we evaluate the performance of the SP-MIR models for the MALP driven by different scenario generation methods through real-world implementation. Runway operations data from HKG and ATL on 31 October 2023 are utilised. The evaluation employs a rolling horizon approach, with decision horizons set at 20-minute intervals for HKG and 10-minute intervals for ATL, as explained in the Appendix B. We take the real-world implementation at HKG as an illustrative example. To manage arriving aircraft, decision-making for the MALP is performed 20 minutes in advance to generate runway assignment plans for the upcoming decision horizon. For instance, if aircraft are scheduled to arrive starting at 12:00, the optimisation process is initiated at 11:40, considering aircraft expected between 12:00 and 12:20. Upon completion, the runway assignment plan for this interval is fixed based on the obtained solution. At 12:00, with updated and more accurate information on arrival and departure times, the sequencing and scheduling plan for aircraft within the 12:00–12:20 window is finalised. Subsequently, a new optimisation run is triggered at 12:00 for the next horizon, covering aircraft scheduled between 12:20 and 12:40. The plan for the previous period must be completed before executing the plan for the next time period. A similar procedure is applied at ATL, with the key difference being the length of the decision horizon, which is set to 10 minutes. This shorter interval is adopted due to the higher volume of arriving aircraft at ATL, necessitating more frequent and responsive decision-making.

We compare deterministic, HDSG, LSG, $OLSG_3$, $OLSG_5$, and $OLSG_{10}$ approaches. For the HDSG approach, the scenario set size $|\Omega|$ is set to 50, whereas for the LSG and all OLSG variants, $|\Omega|$ is set to 100. These parameter settings are selected based on empirical analysis, as they yield the best performance in the actual sample analysis. Table 6 presents the results of the real-world implementation at HKG and ATL, comparing various optimisation approaches across three performance metrics: total objective value, makespan, and environmental cost. At HKG, all SP-MIR models outperform the deterministic benchmark in terms of environmental impact, with $OLSG_5$ achieving the lowest value (94.05), reflecting a reduction of approximately 19.03%. The makespan remains relatively consistent across all methods, ranging from 853.71 to 855.68. A modest decrease in total objective value is also observed, with $OLSG_5$ again delivering the best result (473.88), corresponding to a 2.48% reduction. At ATL, the benefits of SP-MIR models are

Table 6
Results of the real-world implementation..

	HKG			ATL		
	Total	Makespan	Environmental	Total	Makespan	Environmental
Deterministic	485.92	855.68	116.16	382.71	386.35	379.07
HDSG	483.10	855.56	110.64	309.76	364.60	254.92
LSG	483.15	855.18	111.11	302.45	363.01	241.88
$OLSG_3$	479.65	853.72	105.57	306.70	366.39	247.01
$OLSG_5$	473.88	853.71	94.05	284.98	360.99	208.97
$OLSG_{10}$	475.13	854.21	96.04	280.76	356.69	204.83

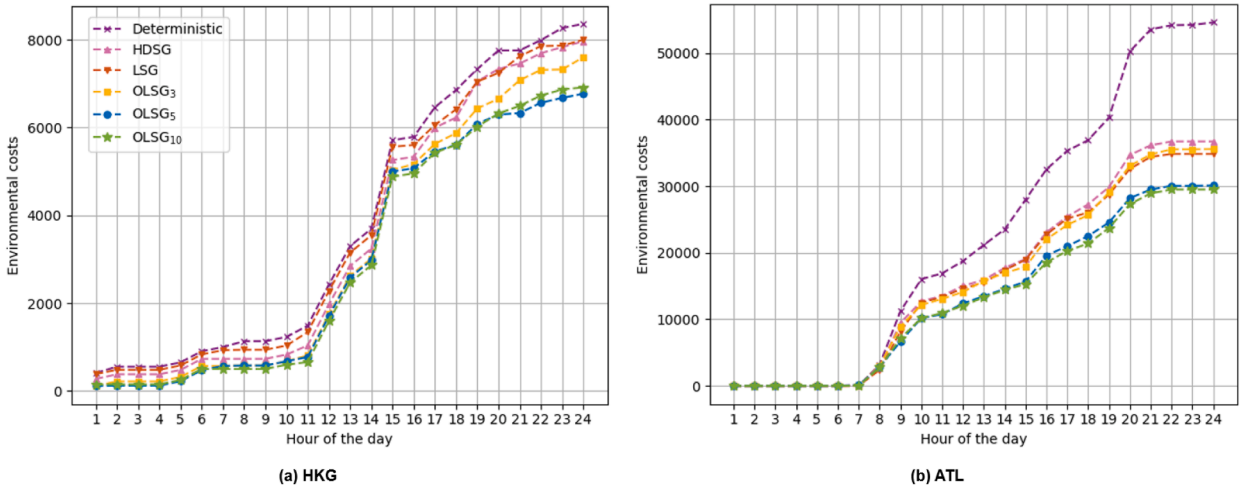


Fig. 6. Hourly cumulative environmental costs for MALP.

particularly pronounced. The deterministic method yields significantly higher total objective value (382.71) and environmental cost (379.07), whereas $OLSG_{10}$ achieves the lowest values in both metrics, 280.76 and 204.83, respectively, corresponding to reductions of 26.6% and 45.9%. Notably, $OLSG_5$ also performs exceptionally well, with total objective value and environmental cost of 284.92 and 211.34, respectively, differing from $OLSG_{10}$ by only 1.5% and 3.2%. The makespan improves slightly across SP-MIR models, with $OLSG_{10}$ recording the shortest duration (356.69), closely followed by $OLSG_5$ (357.84). The superior performance of the OLSG methods can be attributed to the use of a Wasserstein distance-based optimisation approach for selecting representative scenarios. This technique ensures that the sampled scenarios more accurately reflect the true distribution of uncertain parameters, thereby enabling the subsequent SP-MIR model to generate more reliable and efficient multi-runway aircraft landing plans.

In addition, through the line graphs of hourly cumulative environmental costs in Fig. 6, we observe that the $OLSG_5$ and $OLSG_{10}$ methods exhibit lower environmental costs than other methods over time, both at HKG and ATL. 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. Notably, this advantage becomes increasingly pronounced as time progresses.

Similar to the results of the actual sample analysis, the $OLSG_{10}$ method demonstrates the best overall performance in the real-world implementation. However, the performance of $OLSG_5$ is comparable, with only marginal differences across key metrics. Given that the scenario sampling time required by $OLSG_5$ is substantially shorter than that of $OLSG_{10}$, $OLSG_5$ may serve as a more practical and efficient alternative, particularly in time-critical applications such as aircraft landing operations.

6.4. Comparison of SP-MIR and SP-CR models for the 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

Table 7
Performance comparison of SP-CR and SP-MIR models..

	SP-CR			SP-MIR		
	Total	Makespan	Environmental	Total	Makespan	Environmental
HKG	488.87	857.07	120.67	473.88	853.71	94.05
ATL	381.62	383.92	379.32	284.98	360.99	208.97

SP-CR model for the MALP as follows:

$$\min \quad \sum_{\omega\in\Omega} p^{\omega}\left(W_1 z^{\omega} + W_2 \sum_{i\in I} C_i^{\mathrm{Env}} d_i^{\omega}\right) \tag{10a}$$

$$\text{s.t.} \quad (1b),\ (1g),\ (1i)-(1k),\ (1m)-(1o), \tag{10b}$$

$$x_i^r = \sum_{j\in I\cup\{e\}\setminus\{i\}} y_{ij}^r, \quad \forall r\in R, \forall i\in I, \tag{10c}$$

$$\sum_{j\in I} y_{sj}^r = 1, \quad \forall r\in R, \tag{10d}$$

$$\sum_{i\in I} y_{ie}^r = 1, \quad \forall r\in R, \tag{10e}$$

$$\sum_{j\in I\cup\{s\}\setminus\{i\}} y_{ji}^r = \sum_{j\in I\cup\{e\}\setminus\{i\}} y_{ij}^r, \quad \forall r\in R, \forall i\in I, \tag{10f}$$

$$t_i^{\omega} + S_{ij} - t_j^{\omega} \le M^{\omega}(1-y_{ij}^r), \quad \forall r\in R, \forall i\in I, \forall j\in I, i\neq j, \forall\omega\in\Omega, \tag{10g}$$

$$y_{ij}^r \in \{0,1\}, \quad \forall r\in R, \forall i\in I\cup\{s\}, \forall j\in I\cup\{e\}, i\neq j. \tag{10h}$$

Given that the scenarios generated by the OLSG_5 method performed the best in the previous experimental results, we use 100 scenarios generated by the OLSG_5 method when comparing the two SP models. The weights W_1 and W_2 are both set to 0.5. Table 7 presents a comparative evaluation of the SP-CR and SP-MIR models at HKG and ATL, focusing on total objective value, makespan, and environmental cost. The results indicate that the SP-MIR model consistently outperforms the SP-CR model across all metrics. At HKG, the SP-MIR model achieves a total objective value of 473.88, which is a 3.06% reduction compared to the SP-CR model (488.87). The makespan is slightly reduced, decreasing from 857.07 to 853.71. More notably, the environmental cost is reduced by 22.04%, from 120.67 to 94.05, underscoring the SP-MIR model's enhanced sustainability performance. At ATL, the improvements are even more significant. The total objective value drops by 25.33%, from 381.62 to 284.98, while the makespan decreases by 5.98%, from 383.92 to 360.99. The environmental cost shows the most dramatic improvement, with a reduction of 44.91%, from 379.32 to 208.97. These results highlight the SP-MIR model's superior efficiency and environmental responsiveness, particularly in high-intensity runway operations.

Overall, 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.

6.5. 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 impact, respectively, where $W_1 + W_2 = 1$. Fig. 7 illustrates the impact of varying preference levels (W_1, W_2) on both metrics. The results reveal a pronounced sensitivity of environmental impact to preference shifts, while makespan remains largely invariant. As the preference shifts from makespan-focused $(0.99, 0.01)$ to environment-focused $(0.01, 0.99)$, the environmental impact decreases significantly from 23.20 to 17.42 units, corresponding to a reduction of approximately 25.0%. This trend demonstrates the model's adaptability to sustainability-oriented objectives. In contrast, the makespan increases only marginally from 10921.66 to 10922.14, a change of less than 0.005%, indicating that operational efficiency is preserved even under strong environmental prioritisation. The balanced preference scenario $(0.50, 0.50)$ yields intermediate values: a makespan of 10922.04 and an environmental impact of 18.21, offering a practical compromise between throughput and sustainability. In summary, the results highlight a clear trade-off between operational

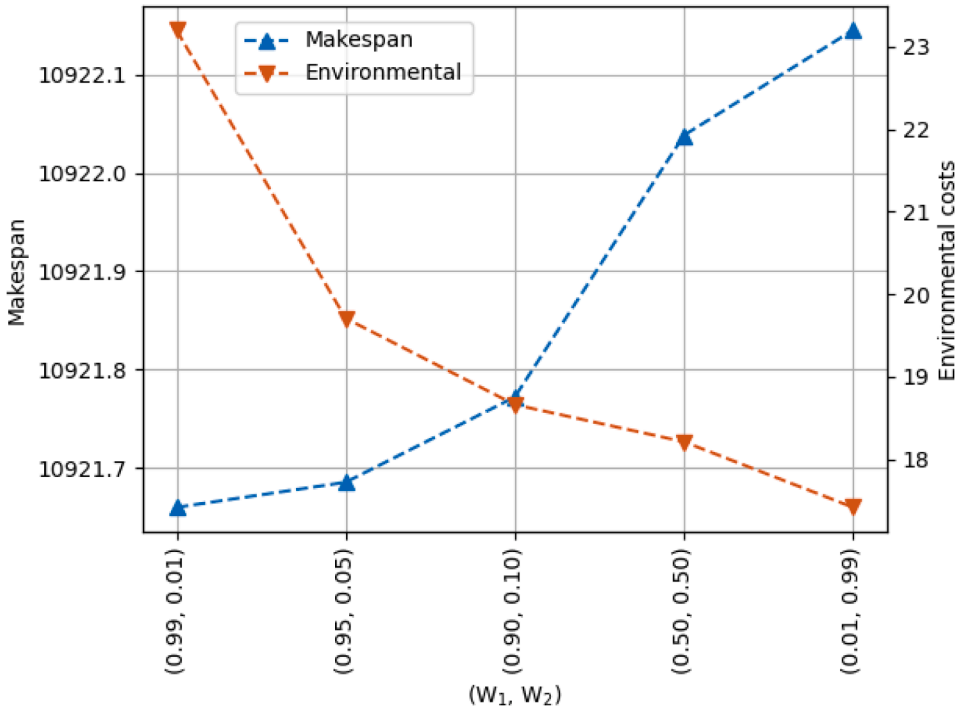


Fig. 7. The impact of preference levels on makespan and environmental costs.

efficiency and environmental performance. By strategically adjusting preference weights, decision-makers can achieve sustainable outcomes without compromising runway operational performance.

6.6. 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, leading 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.

7. Scalability analysis

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 600 seconds. In the scalability analysis, we set the preference for makespan $W_1 = 0.5$ and preference for environmental costs $W_2 = 0.5$. Recall that in the numerical experiments, the $OLSG_5$ method with 100 scenarios exhibited satisfactory performance and required reasonable computa-

Table 8
Comparison of different trust region radius updating schemes in the SBAC method.

	Scheme 1	Scheme 2	Scheme 3
HKG (40)			
Mean Gap (%)	2.54	2.76	2.07
Mean CPU (s)	107.00	104.86	90.38
Mean CPU (s) – MP	12.14	9.92	11.13
Mean CPU (s) – SP	94.86	94.94	79.25
#Non-optimal	3	3	3
ATL (95)			
Mean Gap (%)	8.37	5.08	1.35
Mean CPU (s)	222.58	223.44	173.59
Mean CPU (s) – MP	76.70	72.65	89.40
Mean CPU (s) – SP	145.88	150.79	84.15
#Non-optimal	20	21	14
All (135)			
Mean Gap (%)	6.65	4.40	1.57
Mean CPU (s)	188.34	188.30	148.94
Mean CPU (s) – MP	57.57	54.06	66.21
Mean CPU (s) – SP	130.76	134.24	82.70
#Non-optimal	23	24	17

tional time, underscoring its suitability for real-world applications. Consequently, our scalability analysis focuses on evaluating and comparing the performance of different solution methods when applied to the $OLSG_5$ method with 100 scenarios.

Subsection 7.1 investigates the influence of different trust region radius updating schemes on the performance of the SBAC method. Subsection 7.2 compares the computational efficiency of the proposed SBAC method against established benchmark methods. Finally, Subsection 7.3 presents a comparative analysis of the SBAC method and its enhanced variants, highlighting the benefits of algorithmic enhancements.

7.1. 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 κ is updated to $\kappa \leftarrow \kappa + \lceil \frac{\kappa}{2} \rceil$, where the initial value of κ 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 κ is set to $\lceil 0.1 \times nc \rceil$. After the stabilised master problem becomes infeasible during the convergence process, the value of κ increases successively, taking values of $\kappa \in \{\lceil 0.2 \times nc \rceil, \lceil 0.5 \times nc \rceil, nc\}$. In addition to the aforementioned schemes, we propose a new trust region radius updating strategy designed to provide a more gradual expansion of the solution space. Specifically, the initial radius is set to $\kappa = \lceil 0.2 \times nc \rceil$, then the radius is incrementally increased through the sequence $\kappa \in \{\lceil 0.4 \times nc \rceil, \lceil 0.6 \times nc \rceil, \lceil 0.8 \times nc \rceil, nc\}$. This progressive approach allows the SBAC method to explore the solution space in smaller steps, potentially improving convergence behaviour and solution quality by avoiding sudden expansions. For simplicity, the three schemes mentioned above are denoted as Scheme 1, Scheme 2, and Scheme 3, respectively.

Table 8 presents a comparative evaluation of three trust region radius updating schemes within the SBAC method. This table summarises the computational results using the following notation: “Mean Gap (%)” (mean optimality gap across test instances, calculated as Eq. (11)); “Mean CPU (s)” (mean CPU time in seconds); “Mean CPU (s) - MP” (mean CPU time required by master problem and stabilised master problem in seconds); “Mean CPU (s) - SP” (mean CPU time required by subproblems in seconds); “#Non-optimal” (number of test instances for which the optimal solution was not found).

$$\text{Optimality Gap } (\%) = \frac{\text{UB} - \text{LB}}{\text{UB}} \times 100. \tag{11}$$

As shown in Table 8, Scheme 3 outperforms the other two schemes across most performance metrics. It achieves the lowest overall mean gap of 1.57%, compared to 6.65% and 4.40% for Scheme 1 and Scheme 2, respectively. In terms of computational time, Scheme 3 also demonstrates superior efficiency, with the lowest average CPU time (148.94 seconds), and a significant reduction in subproblem solving time (82.70 seconds), indicating improved convergence behaviour. Moreover, Scheme 3 reduces the number of non-optimal instances to 17, outperforming Scheme 1 (23) and Scheme 2 (24), further confirming its robustness. At the individual airport level, Scheme 3 maintains its advantage. For HKG, it yields the smallest mean gap (2.07%) and the lowest total CPU time (90.38 seconds). For ATL, the performance gap is even more pronounced, with Scheme 3 achieving a mean gap of only 1.35%, compared to 8.37% and 5.08% for the other schemes. These results highlight the effectiveness of the proposed trust region radius updating strategy in Scheme 3, which employs a gradual expansion mechanism. By enabling finer control over the search space, this approach enhances both solution quality and computational efficiency.

Table 9
Performance comparison of the SBAC and benchmark methods.

	LBBD	SLBBD	BAC	SBAC
HKG (40)				
Mean Gap (%)	6.73	8.86	2.05	2.07
Mean CPU (s)	203.89	170.51	99.63	90.38
Mean CPU (s) – MP	160.22	132.92	8.90	11.13
Mean CPU (s) – SP	42.44	37.59	90.73	79.25
#Non-optimal	10	7	3	3
ATL (95)				
Mean Gap (%)	11.77	19.63	8.44	1.35
Mean CPU (s)	327.61	309.19	215.44	173.59
Mean CPU (s) - MP	238.63	271.71	51.16	89.40
Mean CPU (s) - SP	149.52	37.48	164.28	84.15
#Non-optimal	43	42	22	14
All (135)				
Mean Gap (%)	10.27	16.44	6.55	1.57
Mean CPU (s)	290.95	268.10	181.13	148.94
Mean CPU (s) - MP	215.40	230.59	38.64	66.21
Mean CPU (s) - SP	117.79	37.51	142.49	82.70
#Non-optimal	53	49	25	17

7.2. Comparison of the SBAC method and the benchmark solution methods

We adopt the previously proposed LBBD and BAC methods (Elçi and Hooker, 2022; Guo and Zhu, 2023; Li et al., 2023) as benchmark solution methods. Additionally, we directly incorporate the stabilisation strategy introduced by Baena et al. (2020) into the LBBD method, resulting in a variant referred to as stabilised LBBD (SLBBD), which is also included in the benchmark comparison.

Table 9 presents a comprehensive comparison of the proposed SBAC method against three benchmark methods. Across all test instances, SBAC consistently demonstrates superior performance in terms of solution quality and computational efficiency. Specifically, SBAC achieves the lowest mean optimality gap of 1.57% overall, significantly outperforming LBBD (10.27%), SLBBD (16.44%), and BAC (6.55%). This indicates that SBAC yields solutions closer to the optimal on average. In terms of computational time, SBAC also exhibits notable advantages. The total mean CPU time for SBAC is 148.94 seconds, which is substantially lower than LBBD (290.95 seconds), SLBBD (268.10 seconds), and BAC (181.13 seconds). SBAC maintains competitive performance in both master and subproblem phases. Although BAC achieves the fastest master problem CPU time (38.64 seconds), SBAC demonstrates a more efficient overall performance by significantly reducing the subproblem CPU time (82.70 seconds compared to 142.49 seconds for BAC), thereby improving total computational efficiency. Furthermore, SBAC results in the fewest non-optimal instances (17 out of 135), indicating improved robustness and reliability across diverse scenarios. This is particularly evident in the ATL dataset, where SBAC reduces the number of non-optimal cases to 14, compared to 22 for BAC and over 40 for LBBD and SLBBD. In summary, the SBAC method not only enhances solution accuracy but also reduces computational burden, making it a promising and scalable solution method.

7.3. Impact of algorithmic enhancements on the SBAC method

Table 10 presents a comparative analysis of the SBAC method and its enhanced variants, incorporating two algorithmic enhancements: LBLIs and VIs. The inclusion of LBLIs and VIs leads to substantial improvements in both solution quality and computational efficiency. The most enhanced variant, SBAC + LBLIs + VIs, consistently outperforms the baseline SBAC across all metrics and datasets. Specifically, the mean optimality gap is reduced from 1.57% to 0.12% overall, indicating a significant enhancement in solution quality. In terms of CPU time, the improvements of SBAC + LBLIs + VIs are even more pronounced. The total mean CPU time drops from 148.94 seconds to 54.23 seconds, representing a 63.59% reduction. This gain is attributed to accelerated performance in both the master and subproblem phases. The subproblem time, in particular, is reduced from 82.70 seconds to 15.12 seconds, highlighting the effectiveness of the added enhancements in tightening the feasible region and improving convergence. Breaking down the CPU time reveals further insights. For the master problem, the enhancements reduce CPU time from 66.21 seconds (SBAC) to 39.10 seconds (SBAC + LBLIs + VIs), with LBLIs alone achieving the lowest MP time (37.58 seconds). For the subproblem, the CPU time drops from 82.70 seconds to 15.12 seconds in the SBAC + LBLIs + VIs, suggesting that the added inequalities substantially simplify the subproblem structure and accelerate convergence. The number of non-optimal instances also decreases with the enhancements. The baseline SBAC method results in 17 non-optimal test instances, while the SBAC + LBLIs and SBAC + LBLIs + VIs reduce this to just 6, indicating a 64.71% improvement. While LBLIs or VIs alone improve performance compared to the baseline SBAC method, they are less effective than the combined variant (SBAC + LBLIs + VIs). These results suggest that LBLIs and VIs are complementary, and their integration produces a synergistic effect that balances computational effort and enhances solution quality. In summary, incorporating LBLIs and VIs into the SBAC framework yields substantial improvements in both efficiency and quality. The combined variant consistently outperforms the baseline and individual enhancements across all metrics and datasets.

Table 10
Performance evaluation of algorithmic enhancements used in the SBAC method.

	SBAC	SBAC + LBLIs	SBAC + VIs	SBAC + LBLIs + VIs
HKG (40)				
Mean Gap (%)	2.07	**0.00**	1.44	**0.00**
Mean CPU (s)	90.38	9.99	79.40	**7.62**
Mean CPU (s) - MP	11.13	**2.26**	17.21	2.31
Mean CPU (s) - SP	79.25	7.73	62.19	**5.31**
#Non-optimal	3	**0**	2	**0**
ATL (95)				
Mean Gap (%)	1.35	0.24	1.25	**0.17**
Mean CPU (s)	173.59	77.59	164.95	**73.85**
Mean CPU (s) - MP	89.40	**52.46**	100.16	54.60
Mean CPU (s) - SP	84.15	25.13	64.79	**19.25**
#Non-optimal	14	**6**	11	**6**
All (135)				
Mean Gap (%)	1.57	0.17	1.31	**0.12**
Mean CPU (s)	148.94	57.56	139.60	**54.23**
Mean CPU (s) - MP	66.21	**37.58**	75.58	39.10
Mean CPU (s) - SP	82.70	19.97	64.02	**15.12**
#Non-optimal	17	**6**	13	**6**

Table 11
Comparison of SBAC with algorithmic enhancements and commercial solver.

	GUROBI	SBAC + LBLIs + VIs
HKG (40)		
Mean Gap (%)	0.05	**0.00**
Mean CPU (s)	90.07	**7.62**
#Non-optimal	1	**0**
ATL (95)		
Mean Gap (%)	8.36	**0.17**
Mean CPU (s)	291.48	**73.85**
#Non-optimal	34	**6**
All (135)		
Mean Gap (%)	5.90	**0.12**
Mean CPU (s)	231.81	**54.23**
#Non-optimal	35	**6**

Table 11 presents a comparative performance analysis between the commercial solver GUROBI and the SBAC method enhanced with LBLIs and VIs. The enhanced SBAC method demonstrates superior performance in both solution quality and computational efficiency. For the HKG dataset, enhanced SBAC achieves a mean gap of 0.00%, outperforming GUROBI's 0.05%, while reducing the average CPU time from 90.07 seconds to 7.62 seconds. Moreover, enhanced SBAC solves all instances to optimality, whereas GUROBI fails to reach optimality in one instance. In the ATL dataset, which presents greater complexity, the advantages of enhanced SBAC become more pronounced. The mean gap drops from 8.36% (GUROBI) to 0.17% (SBAC + LBLIs + VIs), and the average CPU time is reduced by nearly 74.66%, from 291.48 seconds to 73.85 seconds. Importantly, SBAC yields only six non-optimal solutions, compared to GUROBI's 34. Overall, for all test instances, the enhanced SBAC method has a mean gap of just 0.12% versus GUROBI's 5.90%, and a mean CPU time of 54.23 seconds, which is substantially lower than GUROBI's 231.81 seconds. The number of non-optimal solutions is also reduced from 35 to 6, further confirming the effectiveness of the SBAC with algorithmic enhancements.

8. Conclusions

This paper investigates the SP-MIR model for the MALP under aircraft arrival time uncertainty. This model 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 operational efficiency and environmental sustainability in 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 scenarios and avoiding suboptimal decisions. The numerical results demonstrate that integrating advanced ML-supported scenario generation methods, such as LSG and OLSG, within the SP-MIR model can effectively support the formulation of operationally efficient and environmentally 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-MIR model for MALP, we propose

a novel SBAC method with algorithmic enhancements. This approach decomposes the original problem into an assignment master problem, along with 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 analysis using real data from HKG and ATL. The results show that the SBAC method with algorithmic enhancements significantly outperforms the commercial solver and well-known benchmark methods from the literature.

In the OLSG method proposed in this paper, 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. 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.

CRediT authorship contribution statement

Chenliang Zhang: Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation; **Zhongyi Jin:** Writing – original draft, Visualization, Methodology, Investigation, Data curation; **Kam K. H. Ng:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization; **Ye Liu:** Software, Resources, Project administration, Methodology, Investigation, Data curation; **Lingxiao Wu:** Writing – review & editing, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

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

Acknowledgments

The work described in this paper was supported by grants from Research Grants Council, the Hong Kong Government (Grant No. PolyU25218321, PolyU15201423) and the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong SAR (RJ85, RJJ9, RJ78), the National Natural Science Foundation of China (Grant No. 72301229), and the Research Institute for Sustainable Urban Development (BBG5).

Appendix A. RF method for aircraft arrival time predictions

In this paper, we predict the deviation between the estimated and actual arrival times $\hat{a}_i$ and subsequently add this deviation to the estimated arrival time E_i of aircraft i to determine the predicted arrival time $\hat{E}_i$, which is calculated as $\hat{E}_i = E_i + \hat{a}_i$.

The RF method employed in this study is implemented using the scikit-learn library. The dataset, collected between October 1, and October 31, 2023, comprises 12,822 records from HKG and 33,465 records from ATL, each representing an approaching aircraft and containing relevant operational information. We train separate RF models for HKG and ATL, respectively. Table A.1 shows the 14 features we selected, as well as their data types, encoding methods, and missing value statistics. We find that "Aircraft type", "Cruise speed", "Route distance", and "Fuel load" contain missing values. To resolve this issue, we impute the missing values using the median for "Aircraft type", and the mean for "Cruise speed", 'Route distance", and "Fuel load". Additionally, "Aircraft type", "Domestic/International", and "Wind direction" are represented in a literal format and need to be converted to numerical data. Regarding "Aircraft type", the aircraft at HKG and ATL primarily fall into two categories: heavy and large, which we encode as 2 and 1, respectively. One-hot encoding converts the "Domestic" category to $[1, 0]$ and the "International" category to $[0, 1]$. We consider 16 wind directions and employ one-hot encoding to represent these wind directions. The dataset D comprises data collected from 1st to 30th October 2023. In D, 80% of the data is used as the training set D^{Training}, while the remaining 20% is used as the test set D^{Testing}. The data from 31st October 2023 generates test instances for numerical experiments and scalability analyses.

In the RF method, it is essential to consider and tune several key hyperparameters. We use a grid search method combined with 5-fold cross-validation on D^{Training} to determine the optimal values of key hyperparameters. The tuning results are presented in Table A.2.

We use MAE, mean square error (MSE), and RMSE to evaluate the performance of the RF method. These metrics are defined as follows, with $\bar{E}_i$ and $\hat{E}_i$ representing the actual and predicted arrival times, respectively. The results are provided in Table A.3.

Table A.1
Features of the RF method.

Feature	Data type	Encoding methods	Missing value	
			HKG	ATL
Aircraft type	Object	Label	153	0
Domestic/International	Object	One-hot	0	0
Estimated arrival time	Numerical		0	0
Cruise speed	Numerical		506	64
Straight line distance	Numerical		0	0
Route distance	Numerical		228	223
Estimated flight time	Numerical		0	0
Fuel load	Numerical		820	27
High temperature	Numerical		0	0
Low temperature	Numerical		0	0
Humidity	Numerical		0	0
Barometer	Numerical		0	0
Wind direction	Object	One-hot	0	0
Wind speed	Numerical		0	0

Table A.2
Best hyperparameter values of the RF method.

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

Table A.3
Prediction results on the testing dataset.

	MAE	MSE	RMSE
HKG	21.63	3196.96	56.54
ATL	18.56	2269.12	47.64

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

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

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\bar{E}_i - \hat{E}_i\right)^2}. \tag{A.3}$$

Appendix B. Detailed information of test instances

The test instances are generated based on real-world data collected on 31 October 2023, from HKG and ATL. The number of arriving and departing aircraft at hourly intervals is shown in Fig. B.1. At both HKG and ATL airports, flexible runways are dynamically assigned as either takeoff or landing runways depending on prevailing arrival and departure traffic. During periods of high arrival demand, these flexible runways are primarily utilised for landings to accommodate the increased inbound flow. At such times, HKG operates two runways for arrivals, whereas ATL operates three.

Given the near real-time nature of aircraft sequencing and scheduling decisions, we adopt a decision interval of 10 to 20 minutes to segment the daily data, following the recommendation of Solak et al. (2018). The optimisation frequency for each airport is tailored to its traffic volume and operational complexity. HKG, which accommodates approximately 450 arriving flights per day, is scheduled using a 20-minute decision interval. This choice reflects its relatively moderate arrival density, where the longer planning horizons can be adopted. A less frequent optimisation cycle reduces computational overhead without compromising solution quality, particularly in environments with stable and less congested traffic patterns. In contrast, ATL handles around 900 daily arrivals and requires a more responsive scheduling approach. A 10-minute optimisation interval enables the system to react promptly to high-density arrival

Table B.1
Detailed information of test instances from HKG.

Instance	$\|I\|$	$\|I_H\|$	$\|I_L\|$	$\|R\|$	Instance	$\|I\|$	$\|I_H\|$	$\|I_L\|$	$\|R\|$
HKG_1_1	10	3	7	2	HKG_17_2	7	0	7	2
HKG_1_2	7	1	6	2	HKG_17_3	10	4	6	2
HKG_1_3	4	1	3	2	HKG_18_1	9	2	7	2
HKG_2_1	7	1	6	2	HKG_18_2	7	1	6	2
HKG_2_2	3	1	2	2	HKG_18_3	13	3	10	2
HKG_2_3	5	3	2	2	HKG_19_1	7	3	4	2
HKG_6_2	5	1	4	2	HKG_19_2	11	3	8	2
HKG_6_3	3	1	2	2	HKG_19_3	12	3	9	2
HKG_7_1	5	3	2	2	HKG_20_1	10	2	8	2
HKG_7_3	6	2	4	2	HKG_20_2	10	1	9	2
HKG_8_1	5	3	2	2	HKG_20_3	11	1	10	2
HKG_8_2	4	2	2	2	HKG_21_1	9	1	8	2
HKG_8_3	5	2	3	2	HKG_21_2	8	1	7	2
HKG_11_1	13	2	11	2	HKG_21_3	6	1	5	2
HKG_11_2	6	1	5	2	HKG_23_1	6	2	4	2
HKG_11_3	6	2	4	2	HKG_23_2	7	2	5	2
HKG_16_1	8	2	6	2	HKG_23_3	8	1	7	2
HKG_16_2	7	0	7	2	HKG_24_1	6	0	6	2
HKG_16_3	7	3	4	2	HKG_24_2	7	4	3	2
HKG_17_1	9	5	4	2	HKG_24_3	3	1	2	2

Table B.2
Detailed information of test instances from ATL.

Instance	$\|I\|$	$\|I_H\|$	$\|I_L\|$	$\|R\|$	Instance	$\|I\|$	$\|I_H\|$	$\|I_L\|$	$\|R\|$	Instance	$\|I\|$	$\|I_H\|$	$\|I_L\|$	$\|R\|$
ATL_6_5	4	0	4	2	ATL_13_1	12	0	12	3	ATL_18_3	7	0	7	3
ATL_7_1	4	2	2	3	ATL_13_2	9	0	9	3	ATL_18_4	6	1	5	3
ATL_7_6	8	0	8	3	ATL_13_3	8	0	8	3	ATL_18_5	7	1	7	3
ATL_8_1	7	0	7	3	ATL_13_4	7	0	7	3	ATL_18_6	12	0	12	3
ATL_8_2	15	1	14	3	ATL_13_5	12	0	12	3	ATL_19_1	13	0	13	3
ATL_8_3	12	1	11	3	ATL_13_6	5	0	5	3	ATL_19_2	10	1	9	3
ATL_8_4	12	0	12	3	ATL_14_1	5	0	5	3	ATL_19_3	14	0	14	3
ATL_8_5	9	2	7	3	ATL_14_2	8	0	8	3	ATL_19_4	9	1	8	3
ATL_8_6	12	0	12	3	ATL_14_3	10	0	10	3	ATL_19_5	11	2	9	3
ATL_9_1	14	0	14	3	ATL_14_4	11	0	11	3	ATL_19_6	7	1	6	3
ATL_9_2	15	1	14	3	ATL_14_5	12	0	12	3	ATL_20_1	11	0	11	3
ATL_9_3	16	0	16	3	ATL_14_6	10	0	10	3	ATL_20_2	12	0	12	3
ATL_9_4	15	0	15	3	ATL_15_1	7	0	7	3	ATL_20_3	14	1	13	3
ATL_9_5	15	0	15	3	ATL_15_2	9	1	8	3	ATL_20_4	15	0	15	3
ATL_9_6	8	0	8	3	ATL_15_3	12	1	11	3	ATL_20_5	15	1	14	3
ATL_10_1	11	0	11	2	ATL_15_4	8	0	8	3	ATL_20_6	11	0	11	3
ATL_10_2	7	0	7	2	ATL_15_5	8	0	8	3	ATL_21_1	9	0	9	3
ATL_10_3	13	0	13	2	ATL_15_6	6	1	5	3	ATL_21_2	5	0	5	3
ATL_10_4	8	0	8	2	ATL_16_1	7	0	7	2	ATL_21_3	11	0	11	3
ATL_10_5	10	0	10	2	ATL_16_2	10	0	10	2	ATL_21_4	13	3	10	3
ATL_10_6	10	0	10	2	ATL_16_3	11	2	9	2	ATL_21_5	6	1	5	3
ATL_11_1	6	1	5	2	ATL_16_4	11	1	10	2	ATL_21_6	8	0	8	3
ATL_11_2	9	0	9	2	ATL_16_5	9	0	9	2	ATL_22_2	7	2	5	2
ATL_11_3	6	0	6	2	ATL_16_6	8	0	8	2	ATL_22_3	8	1	7	2
ATL_11_5	3	0	3	2	ATL_17_1	8	2	6	3	ATL_22_4	5	0	5	2
ATL_11_6	7	0	7	2	ATL_17_2	9	1	8	3	ATL_22_5	4	0	4	2
ATL_12_1	8	0	8	2	ATL_17_3	11	1	10	3	ATL_23_2	3	0	3	2
ATL_12_2	7	0	7	2	ATL_17_4	10	0	10	3	ATL_23_4	5	0	5	2
ATL_12_3	5	0	5	2	ATL_17_5	14	1	13	3	ATL_24_2	5	0	5	2
ATL_12_4	3	0	3	2	ATL_17_6	13	0	13	3	ATL_24_3	3	0	3	2
ATL_12_5	10	0	10	2	ATL_18_1	10	0	10	3	ATL_24_4	3	0	3	2
ATL_12_6	6	0	6	2	ATL_18_2	10	0	10	3					

streams and frequent fluctuations in traffic conditions. This design ensures that each airport is managed with an appropriate level of temporal granularity, balancing operational responsiveness with computational efficiency.

After excluding test instances in which the number of runways used for landing is less than or equal to one, or the number of arrival aircraft is fewer than the number of available runways, a total of 135 test instances remain, with 40 corresponding to HKG and 95 to ATL. Table B.1 presents detailed information on the test instances derived from HKG. The instance code HKG_1_1 corresponds to the first hour of the selected operational day, with the first 20-minute subinterval considered within this time frame. For each test instance, we report the total number of arriving aircraft ($|I|$), including the number of heavy ($|I_H|$) and large ($|I_L|$) aircraft, as well

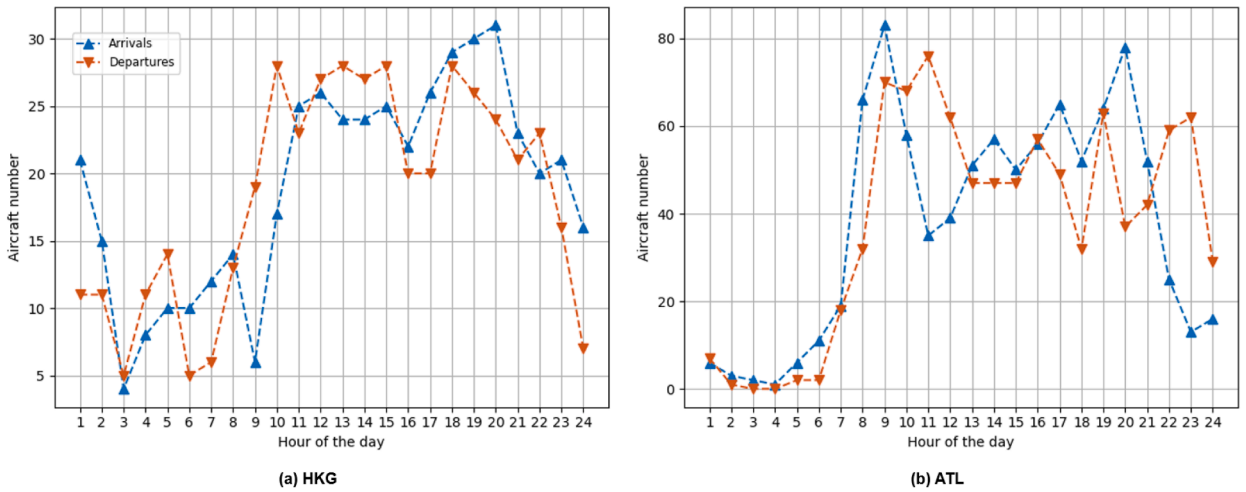


Fig. B.1. Hourly number of aircraft arrivals and departures at HKG and ATL on 31 October 2023.

as the number of runways available for landing operations ($|R|$). Table B.2 presents detailed information on the test instances derived from ATL, which follow the same naming convention.

Appendix C. Performance evaluation of OLSG using out-of-sample data

In this appendix, we conduct an out-of-sample analysis to evaluate how well OLSG preserves the statistical characteristics of the uncertain parameters. We conduct out-of-sample analysis on the LSG and OLSG methods with a scenario size of 100. First, the optimal solution $\hat{\mathbf{x}}^*$ provided by the SP-MIR model is fixed. Then, we solve the second-stage problem $\mathbb{E}_{\omega \in \Omega_{\text{out}}}\left[Q(\hat{\mathbf{x}}^*, \omega)\right]$ with an out-of-sample scenario set Ω_{out} of size 1,000, which is generated in the same way as the scenario set Ω.

Table C.1 compares the out-of-sample performance of the baseline LSG method and three OLSG variants with selection ratios $\gamma = 3$, 5, and 10. All methods yield nearly identical makespan values, indicating that the scheduling efficiency is preserved across different scenario generation strategies. However, notable differences are observed in the environmental cost across the methods in the out-of-sample analysis. As the selection ratio γ increases, the environmental cost gradually decreases, with values ranging from 30.19 under the LSG method to 29.21 under OLSG_{10}. This trend suggests that OLSG, by selecting more representative scenarios, better captures the underlying distributional patterns of the uncertain parameters. The improved environmental performance on the out-of-sample set Ωout confirms that OLSG better preserves the statistical characteristics of the uncertainty. These results support the rationale that OLSG improves model performance by solving a clustering problem that yields more representative scenario subsets. Although the absolute reduction in environmental cost ranges from 2% to 3%, such improvements are considered meaningful in stochastic optimisation. In real-world applications, particularly in the aviation industry, runway scheduling decisions directly influence fuel burn and emissions due to aircraft idling and taxiing delays. Even modest percentage reductions in environmental impact can lead to significant long-term benefits when aggregated over thousands of daily operations. These improvements reinforce the practical value of representative scenario selection, as achieved by the OLSG method, which better captures the uncertainty in traffic patterns and operational disruptions.

Table C.1
Out-of-sample analysis.

	LSG	OLSG_3	OLSG_5	OLSG_{10}
Makespan	10212.43	10212.54	10212.10	10212.59
Environmental cost	30.19	29.50	29.27	29.21

References

Adulyasak, Y., Cordeau, J.-F., Jans, R., 2015. Benders decomposition for production routing under demand uncertainty. Oper. Res. 63 (4), 851–867.
Baena, D., Castro, J., Frangioni, A., 2020. Stabilized benders methods for large-scale combinatorial optimization, with application to data privacy. Manage. Sci. 66 (7), 3051–3068.
Balakrishnan, H., Chandran, B.G., 2010. Algorithms for scheduling runway operations under constrained position shifting. Oper. Res. 58 (6), 1650–1665.
Ban, G.-Y., Rudin, C., 2019. The big data newsvendor: practical insights from machine learning. Oper. Res. 67 (1), 90–108.
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. Springer, pp. 84–98.

Bennell, J.A., Mesgarpour, M., Potts, C.N., 2011. Airport runway scheduling. 4OR 9, 115–138.
Bennell, J.A., Mesgarpour, M., Potts, C.N., 2017. Dynamic scheduling of aircraft landings. Eur. J. Oper. Res. 258 (1), 315–327.
Bertsimas, D., Kallus, N., 2020. From predictive to prescriptive analytics. Manage. Sci. 66 (3), 1025–1044.
Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.
Chen, S., Wu, L., Ng, K. K.H., Liu, W., Wang, K., 2024. How airports enhance the environmental sustainability of operations: a critical review from the perspective of operations research. Transp. Res. Part E: Logist. Transp. Rev. 183, 103440.
Elçi, Ö., Hooker, J., 2022. Stochastic planning and scheduling with logic-based benders decomposition. INFORMS J. Comput. 34 (5), 2428–2442.
Fachini, R.F., Armentano, V.A., 2020. Logic-based benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. Comp. Indust. Eng. 148, 106641.
Fischetti, M., Lodi, A., 2003. Local branching. Math. Program. 98, 23–47.
Gong, H., Zhang, Z.-H., 2022. Benders decomposition for the distributionally robust optimization of pricing and reverse logistics network design in remanufacturing systems. Eur. J. Oper. Res. 297 (2), 496–510.
Guo, P., Zhu, J., 2023. Capacity reservation for humanitarian relief: a logic-based benders decomposition method with subgradient cut. Eur. J. Oper. Res. 311 (3), 942–970.
Harikiopoulo, D., Neogi, N., 2010. Polynomial-time feasibility condition for multiclass aircraft sequencing on a single-runway airport. IEEE Trans. Intell. Transp. Syst. 12 (1), 2–14.
Heidt, A., Helmke, H., Kapolke, M., Liers, F., Martin, A., 2016. Robust runway scheduling under uncertain conditions. J. Air Transp. Manage. 56, 28–37.
Hochreiter, R., Pflug, G.C., 2007. Financial scenario generation for stochastic multi-stage decision processes as facility location problems. Ann. Oper. Res. 152 (1), 257–272.
Hong, Y., Choi, B., Kim, Y., 2018. Two-stage stochastic programming based on particle swarm optimization for aircraft sequencing and scheduling. IEEE Trans. Intell. Transp. Syst. 20 (4), 1365–1377.
Hong, Y., Choi, B., Lee, K., Kim, Y., 2017. Dynamic robust sequencing and scheduling under uncertainty for the point merge system in terminal airspace. IEEE Trans. Intell. Transp. Syst. 19 (9), 2933–2943.
Hooker, J.N., 2007. Planning and scheduling by logic-based benders decomposition. Oper. Res. 55 (3), 588–602.
Ikli, S., Mancel, C., Mongeau, M., Olive, X., Rachelson, E., 2021. The aircraft runway scheduling problem: a survey. Comp. Oper. Res. 132, 105336.
Kapolke, M., Fürstenau, N., Heidt, A., Liers, F., Mittendorf, M., Weiß, C., 2016. Pre-tactical optimization of runway utilization under uncertainty. J. Air Transp. Manage. 56, 48–56.
Khassiba, A., Bastin, F., Cafieri, S., Gendron, B., Mongeau, M., 2020. Two-stage stochastic mixed-integer programming with chance constraints for extended aircraft arrival management. Transp. Sci. 54 (4), 897–919.
Khassiba, A., Cafieri, S., Bastin, F., Mongeau, M., Gendron, B., 2022. Two-stage stochastic programming models for the extended aircraft arrival management problem with multiple pre-scheduling points. Transp. Res. Part C: Emerg. Technol. 142, 103769.
Kleywegt, A.J., Shapiro, A., Homem-de Mello, T., 2002. The sample average approximation method for stochastic discrete optimization. SIAM J. Optim. 12 (2), 479–502.
Laporte, G., Louveaux, F.V., 1993. The integer l-shaped method for stochastic integer programs with complete recourse. Oper. Res. Lett. 13 (3), 133–142.
Li, Y., Côté, J.-F., Coelho, L.C., Zhang, C., Zhang, S., 2023. Order assignment and scheduling under processing and distribution time uncertainty. Eur. J. Oper. Res. 305 (1), 148–163.
Lieder, A., Briskorn, D., Stolletz, R., 2015. A dynamic programming approach for the aircraft landing problem with aircraft classes. Eur. J. Oper. Res. 243 (1), 61–69.
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. Appl. Soft. Comput. 98, 106853.
Ng, K. K.H., Lee, C. K.M., Chan, F. T.S., Chen, C.-H., Qin, Y., 2020. A two-stage robust optimisation for terminal traffic flow problem. Appl. Soft. Comput. 89, 106048.
Ng, K. K.H., Lee, C. K.M., Chan, F. T.S., Qin, Y., 2017. Robust aircraft sequencing and scheduling problem with arrival/departure delay using the min-max regret approach. Transp. Res. Part E: Logist. Transp. Rev. 106, 115–136.
Pohl, M., Kolisch, R., Schiffer, M., 2021. Runway scheduling during winter operations. Omega 102, 102325.
Qi, M., 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. INFORMS, pp. 36–58.
Rahimian, H., Pagnoncelli, B., 2023. Data-driven approximation of contextual chance-constrained stochastic programs. SIAM J. Optim. 33 (3), 2248–2274.
Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W., 2017. The benders decomposition algorithm: a literature review. Eur. J. Oper. Res. 259 (3), 801–817.
Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W., 2018. Accelerating the benders decomposition method: application to stochastic network design problems. SIAM J. Optim. 28 (1), 875–903.
Reese, J., 2006. Solution methods for the p-median problem: an annotated bibliography. NETWORKS: Int. J. 48 (3), 125–142.
Sabar, N.R., 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.
Sadana, U., Chenreddy, A., Delage, E., Forel, A., Frejinger, E., Vidal, T., 2025. A survey of contextual optimization methods for decision-making under uncertainty. Eur. J. Oper. Res. 320 (2), 271–289.
Salehipour, A., 2020. An algorithm for single-and multiple-runway aircraft landing problem. Math. Comput. Simul. 175, 179–191.
Serhan, D., Lee, H., Yoon, S.W., 2018. Minimizing airline and passenger delay cost in airport surface and terminal airspace operations. J. Air Transp. Manage. 73, 120–133.
Solak, S., Solveling, G., Clarke, J.-P.B., Johnson, E.L., 2018. Stochastic runway scheduling. Transp. Sci. 52 (4), 917–940.
Sölveling, G., Solak, S., Clarke, J.-P., Johnson, E., 2011a. Runway operations optimization in the presence of uncertainties. J. Guidance, Control, Dynam. 34 (5), 1373–1382.
Sölveling, G., Solak, S., Clarke, J.-P.B., Johnson, E.L., 2011b. Scheduling of runway operations for reduced environmental impact. Transp. Res. Part D: Transp. Environ. 16 (2), 110–120.
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. Springer, pp. 16–30.
Tian, X., Yan, R., Liu, Y., Wang, S., 2023a. A smart predict-then-optimize method for targeted and cost-effective maritime transportation. Transp. Res. Part B: Methodol. 172, 32–52.
Tian, X., Yan, R., Wang, S., Laporte, G., 2023b. Prescriptive analytics for a maritime routing problem. Ocean Coastal Manage. 242, 106695.
Tian, X., Yan, R., Wang, S., Liu, Y., Zhen, L., 2023c. Tutorial on prescriptive analytics for logistics: what to predict and how to predict. Elect. Res. Arch. .
Tian, Y., Wan, L., Han, K., Ye, B., 2018. Optimization of terminal airspace operation with environmental considerations. Transp. Res. Part D: Transp. Environ. 63, 872–889.
Tran, T.T., Araujo, A., Beck, J.C., 2016. Decomposition methods for the parallel machine scheduling problem with setups. INFORMS J. Comput. 28 (1), 83–95.
Van Parys, B. P.G., Esfahani, P.M., Kuhn, D., 2021. From data to decisions: distributionally robust optimization is optimal. Manage. Sci. 67 (6), 3387–3402.
Wang, H., Sun, Q., Wang, S., 2025. Data-driven models for optimizing second-hand ship trading strategies under contextual information. NRL 72 (2), 275–291.
Wang, K., Jacquillat, A., 2020. A stochastic integer programming approach to air traffic scheduling and operations. Oper. Res. 68 (5), 1375–1402.
Wang, S., Yan, R., 2023. Fundamental challenge and solution methods in prescriptive analytics for freight transportation. Transp. Res. Part E: Logist. Transp. Rev. 169, 102966.
Wu, L., Adulyasak, Y., Cordeau, J.-F., Wang, S., 2022. Vessel service planning in seaports. Oper. Res. 70 (4), 2032–2053.
Yan, R., Yang, Y., Du, Y., 2022. Stochastic optimization model for ship inspection planning under uncertainty in maritime transportation. Elect. Res. Arch. .
Yang, Y., Yan, R., Wang, S., 2024. Prescriptive analytics models for vessel inspection planning in maritime transportation. Comp. Indust. Eng. 190, 110012.

Zhang, C., Jin, Z., Ng, K. K.H., Tang, T.-Q., Tang, R., 2025a. Distributionally robust optimisation approach for aircraft sequencing and scheduling with learning-driven arrival and departure time predictions. Omega , 103415.
Zhang, C., Jin, Z., Ng, K. K.H., Tang, T.-Q., Zhang, F., Liu, W., 2025b. Predictive and prescriptive analytics for robust airport gate assignment planning in airside operations under uncertainty. Transp. Res. Part E: Logist. Transp. Rev. 195, 103963.
Zhang, W., Wang, K., Jacquillat, A., Wang, S., 2023. Optimized scenario reduction: solving large-scale stochastic programs with quality guarantees. INFORMS J. Comput. 35 (4), 886–908.