

Article

Stochastic Departure Metering with Machine-Learning Taxi-Time Predictors

Chang Liu ¹ and Eri Itoh ^{1,2,*} 

¹ Department of Aeronautics and Astronautics, The University of Tokyo, Tokyo 113-8656, Japan; changliu@g.ecc.u-tokyo.ac.jp

² Research Center for Advanced Science and Technology, The University of Tokyo, Tokyo 153-8904, Japan

* Correspondence: eriitoh@g.ecc.u-tokyo.ac.jp

Abstract

In busy airports, departure metering (DM) can reduce unnecessary runway-area queueing by holding aircraft at their stands, but its effectiveness is highly sensitive to the accurate estimation of taxi time. Existing studies generally pursue two separate approaches to improve the effectiveness of DM decisions: (1) improving the accuracy of taxi-time prediction, and (2) incorporating taxi-time uncertainty to make more robust decisions. However, prediction errors remain unavoidable, and uncertainty sets constructed directly from historical data of multiple flights fail to capture the heterogeneity between flights, so the previous approaches always result in suboptimal decisions. Thus, this research develops a learning-based two-stage stochastic DM that constructs the uncertainty sets from the outputs of machine-learning predictors and makes robust decisions under uncertainty. This framework can capture the uncertainty set for each flight more accurately and effectively reduce the influence of prediction errors on DM decisions. The DM model is formulated as a two-stage stochastic program integrating first-stage pushback decisions and second-stage runway sequencing, solved by the sample average approximation and scenario generation method. The proposed framework is evaluated using real-world data from Tokyo International Airport (RJTT). The results show that the framework can effectively reduce the overall queueing level and right-tail risk of runway-area queueing time compared with the certainty model, with reduced expected queueing time during peak hours. Additional equity and sensitivity analyses further show that these improvements are realized without being achieved at the expense of a small subset of flights and remain stable under schedule perturbations. This study demonstrates that learning-based uncertainty representations can enhance the robustness and operational quality of DM decisions.

Keywords: airport surface operations; departure metering; taxi-time prediction; stochastic programming; learning-based optimization



Academic Editor: Weili Zeng

Received: 18 June 2026

Revised: 23 July 2026

Accepted: 27 July 2026

Published: 29 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

At busy hub airports, limited runway capacity and dense surface traffic interactions often lead to congestion and queueing near the runway entrance area before takeoff. Statistical data from Tokyo International Airport (RJTT) in 2024 show that about 17–19% of departures experience more than 3 min of runway queueing and 6–7% exceed 5 min during peak hours (10:00–12:00) every day. This phenomenon is primarily caused by the imbalance between flight demand and runway service capacity. Under limited expansion of airport infrastructure in the short term, high-efficiency operational strategies are needed

to mitigate runway queueing. Among these operational strategies, departure metering (DM) has been widely adopted to improve the efficiency of airport surface operations. DM is essentially a pushback-scheduling optimization problem. By delaying the time of aircraft pushback from the spot, DM aims to balance surface demand and runway service capacity, thereby reducing unnecessary queueing near the runway. The effectiveness of DM critically depends on the ability to estimate the taxi time from pushback to the runway area. If taxi time is misestimated, DM may optimize in the wrong direction, leading not only to suboptimal decisions but also potentially to increased delays. Early DM studies typically relied on taxi-time estimates derived from historical operational data and formulated the problem as a certainty optimization model, without explicitly accounting for taxi-time uncertainty. Considering the sensitivity of DM to taxi-time errors, current studies try to improve DM from two aspects: improving taxi-time prediction accuracy to reduce uncertainty and developing uncertainty-aware optimization models to make more robust decisions under uncertainty.

From the first aspect, current research mainly resorts to data-driven machine-learning methods to improve taxi-time prediction accuracy [1]. Prior studies have shown that such methods can significantly improve prediction accuracy [2–6] and can support decision making in airport surface operations [1]. However, even with advanced prediction models, taxi time remains inherently uncertain due to complex and partially unobservable operational factors, such as local traffic interactions, controller instructions, and stochastic disturbances. From the second aspect, researchers try to capture operational uncertainties and formulate uncertainty DM optimization models [7–10] using robust optimization operation [7] and stochastic programming [9,11] to model the problem. Although these approaches lead to more reliable decisions, limitations remain: existing uncertainty representations are often constructed from historical data, and multiple or even all flights may share the same uncertainty structure, which may not fully capture the heterogeneity between flights. Under this setting, the flights with little uncertainty may be over-protected, while those with large uncertainty may be under-protected, leading to suboptimal decisions.

Therefore, although existing studies have reduced the impact of taxi-time estimation errors on DM from two different perspectives, each research direction still suffers from inherent limitations when applied independently. Machine-learning methods improve taxi-time estimation accuracy, but the remaining prediction uncertainty is not explicitly considered during optimization. Uncertainty optimization explicitly accounts for taxi time uncertainty in decision-making, but existing uncertainty representations are often constructed from historical statistics and cannot adequately capture flight-specific uncertainty. Therefore, a critical research gap remains in developing an effective framework that bridges machine-learning taxi-time prediction and uncertainty optimization, enabling predictive uncertainty to be systematically propagated into DM optimization.

However, implementing such a framework presents two main challenges: selecting informative features without imposing excessive computational costs when integrating ML with DM, and formulating an appropriate representation of taxi-time uncertainty. Although ML prediction and uncertainty modelling are considered simultaneously in Kang et al. [11], the inappropriate model formulation and excessive simplification fail to capture the flight queueing influence under different taxi-time realizations.

This research, therefore, develops a learning-based two-stage stochastic programming (SP) framework for DM. First, a machine-learning prediction model is trained using historical data. For each operational instance, the trained model predicts flight-level taxi times and their distributions. The resulting predictions are then used to formulate a two-stage SP model that integrates here-and-now pushback decisions in the first stage and wait-and-see runway sequencing in the second stage. The cost of the runway sequencing, runway

queueing time, is regarded as the recourse for DM decisions. The proposed framework is evaluated using real-world data from Tokyo International Airport (RJTT), which serves as a representative case study to demonstrate the applicability and effectiveness of the proposed methodology. The results show that the framework can effectively reduce the overall queueing level and right-tail risk of runway-area queueing time compared with the certainty model, reducing expected queueing time during peak hours. Additional analyses further show that these improvements are realized without being achieved at the expense of a small subset of flights and remain stable under schedule perturbations, supporting the practical applicability of the proposed framework. This study demonstrates that learning-based uncertainty representations can enhance the robustness and operational quality of DM decisions.

The main contributions of this study are as follows.

- This study develops a two-stage stochastic DM framework that integrates first-stage pushback decisions with second-stage runway sequencing under realized taxiing conditions. By explicitly accounting for recourse across taxi-time scenarios, the framework supports more robust DM decisions than a certainty-model formulation.
- This study connects ML-based prediction with stochastic optimization through flight-level taxi-time distributions. This provides a more informative uncertainty representation than conventional statistical approaches that impose a common uncertainty structure across multiple flights.
- The evaluation using real-world data from Tokyo International Airport (RJTT) demonstrates that the proposed stochastic framework reduces both the overall level and right-tail risk of runway-area queueing compared with the certainty model. Comparisons between distribution construction approaches further show the value of accurately representing uncertainty, while the equity and sensitivity analyses support the operational applicability of the resulting decisions.
- This study compares static and dynamic feature sets for taxi-time prediction and assesses their suitability as inputs to DM optimization. The results show that static-feature-based prediction models retain strong predictive performance while avoiding the circular dependency between dynamic traffic-state features and pushback decisions, enabling a practical sequential integration of prediction and optimization.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on departure metering and taxi-time prediction. Section 3 presents the methodology of the proposed learning-based two-stage SP framework containing taxi-time prediction model training, distribution construction, and two-stage SP model formulation and solution. Section 4 presents numerical experiments based on real data from Tokyo International Airport (RJTT), and Section 5 concludes the paper.

2. Related Work

The literature related to this study can be organized into two main aspects: departure metering (DM) and taxi-time prediction. The first aspect covers decision-support and optimization methods that determine gate-holding, pushback release, metering policies, or departure schedules. The second aspect examines how taxi time and runway-arrival time can be estimated from operational data, including both prediction models and analytical models of the departure process. These two aspects are closely connected because the effectiveness of DM depends on accurate and available taxi-time information.

2.1. Departure Metering

The operational procedures of a departure flight are shown in Figure 1. Note that Figure 1 is intended only as a conceptual illustration of the departure process. The terminal,

taxiway, and runway shown in the figure do not represent the actual airport layout, and only a single runway is illustrated for simplicity. Initially, a departure flight is prepared at the gate or spot, where all required pre-departure ground handling activities are completed, including passenger boarding, baggage loading, catering, aircraft fueling by fuel dispenser trucks, and pushback vehicle preparation. Once these activities have been completed and the aircraft is ready for pushback, the pilot requests pushback clearance from the ground controller. The ground controller then decides whether to approve the pushback request immediately or delay it to a later time. Once the pushback is approved, the flight begins the pushback procedure. After completing pushback, the flight taxis to the runway area and queues in the runway area until it receives takeoff clearance. Based on the estimated time at which each flight becomes ready for pushback, departure metering (DM) determines an appropriate pushback time for each departure flight to balance surface demand and runway service capacity, thereby reducing unnecessary runway queuing.

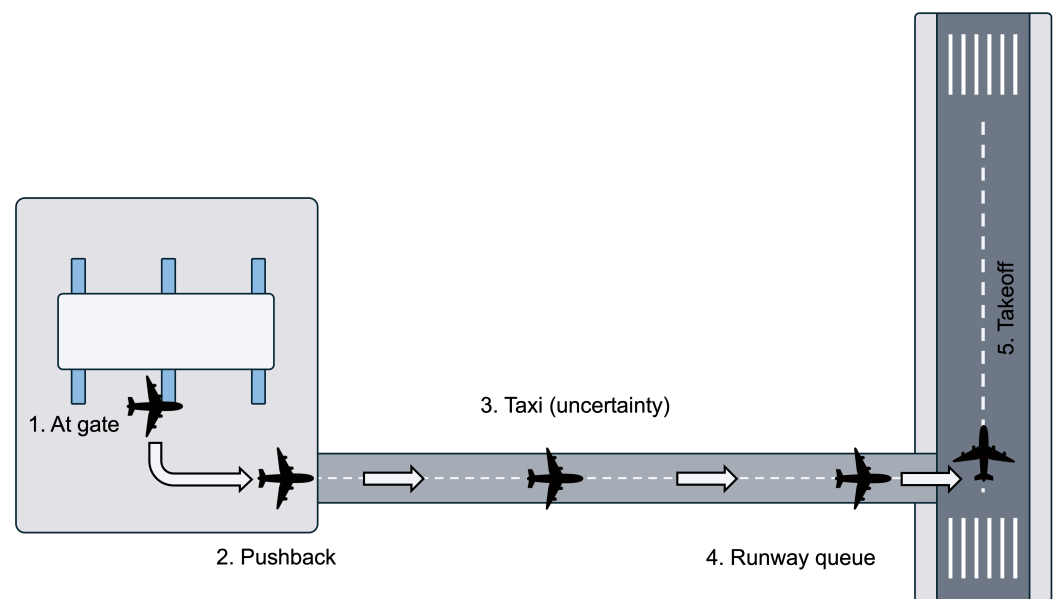


Figure 1. A conceptual illustration of the general operational procedures of a departure flight.

Existing DM studies can generally be classified into several categories. One important class of DM methods is aggregate pushback-rate control. Simaiakis et al. [12] designed and field-tested pushback rate control at Boston Logan Airport, showing that gate holding can reduce taxi-out time and fuel burn without sacrificing practical implementability. Related work developed decision-support tools for pushback rate control and evaluated congestion management in reality [13,14]. These studies determine release rates or gate-hold actions at the aggregate level, which makes them operationally simple but less flight-specific.

Another line of DM research focuses on individual aircraft rather than aggregate pushback rates. Individual-level approaches determine aircraft-specific release times, takeoff sequences, or runway schedules subject to separation constraints and operational restrictions. Malik et al. [15,16] investigated departure aircraft release decisions for efficient surface operations, and related NASA surface-management studies developed optimization-based decision support for runway scheduling and spot release planning. Bertsimas and Frankovich [17] proposed a unified airport traffic-flow optimization framework, showing how surface and runway decisions can be formulated in an integer optimization setting. Such individual-level models are attractive because they can explicitly account for flight heterogeneity in operations. However, they rely on reliable parameter estimates to accurately characterize each flight's future spatiotemporal operational states, such as the time at which an aircraft is expected to reach the runway area after pushback.

Given the presence of various sources of uncertainty in airport surface operations, certainty-model DM solutions may be undermined by parameter misspecification, leading to suboptimal decisions in practice. Several studies, therefore, explicitly account for uncertainties in DM. Murça [7] formulated airport DM as a robust optimization problem under uncertain taxi-out time predictions and showed that robustness can improve takeoff-time predictability and reduce taxi-out delays. Badrinath et al. [8] examined the earliest off-block-time (pushback readiness time) uncertainty and evaluated its influence on DM decisions, showing that such uncertainty can substantially undermine the benefits of DM. Chen and Solak [9] further considered uncertainty in the number of arrivals in addition to the earliest off-block time and developed a stochastic dynamic optimization model for DM. Chandra et al. [10] extended DM by considering uncertainties in taxi time and standard instrument departure (SID) track time and tried to minimize en-route conflict risk. These studies provided more robust DM decisions, but they usually constructed uncertainty sets directly from historical data by statistical methods, assuming that multiple flights share the same uncertainty structure. Such an approach may fail to capture the heterogeneity between flights, leading to suboptimal decisions.

In addition to uncertainty-aware DM models, recent studies have also explored simulation-based and learning-based approaches to DM. These studies aim to improve the accuracy of parameter estimation so that the optimization model can identify more suitable solutions without incurring additional conservatism in uncertainty models. Badrinath et al. [18] compared DM at USA and European airports, highlighting how airport layout, procedures, and uncertainty influence the realizable benefit of metering. Kato and Itoh [1] combined ML-based taxi-time prediction and a simple DM model, and the simulation results showed that the improvements in taxi-time prediction accuracy brought by ML models can significantly mitigate queueing and delays. In addition, deep reinforcement learning has also been used to learn pushback or metering policies under spatiotemporal airside interactions [19,20]. However, the parameters can still be misestimated, and the uncertainty is not explicitly modeled in these studies, which may lead to suboptimal decisions when prediction errors are large.

Most similar to this research, Kang et al. [11] integrated data-driven taxi-time uncertainty prediction with a two-stage SP framework for DM. While the study provides a learning-based approach to incorporate flight-level taxi-time uncertainty into DM, its treatment of uncertainty remains limited. Although formulated as a two-stage stochastic program, the uncertainty is completely absorbed into the first-stage constraints via a quantile-based reformulation, which reduces the problem to a certainty-model approximation and does not explicitly account for second-stage recourse costs under different realizations. In addition, modeling taxi-time uncertainty through chance constraints is questionable, as runway arrival time should be determined by pushback time and realized taxi time rather than relaxed as an inequality. More importantly, departure system performance depends on the temporal interactions among aircraft, whereas marginal taxi-time quantiles fail to capture critical scenarios such as near-simultaneous arrivals that lead to queue formation, potentially resulting in a misleading notion of robustness.

In summary, the DM literature has progressed from aggregate release-rate control to flight-level optimization, and from certainty models to robust, stochastic, and learning-based approaches. However, there remains a practical gap in how taxi-time predictions should be incorporated into a stochastic DM model considering the trade-off between prediction accuracy and model computational cost. This research focuses on a two-stage SP framework that integrates the first-stage pushback decision before taxi-time realization and the second-stage runway sequencing decision that reacts to realized taxiing conditions.

This setting makes the quality, uncertainty representation, and time of taxi-time prediction central to the optimization model.

2.2. Taxi-Time Prediction

Taxi-time prediction has been studied for several decades because it plays an important role in surface operations (see [2] and references therein). Considering that taxi time is strongly affected by surface congestion and runway queues [21,22], early work separated taxi-out into unimpeded taxi and runway-queueing components and derived queueing time using analytical queueing models [23,24]. These methods, while capable of capturing key operational mechanisms, usually require sophisticated and system-specific implementation. Moreover, the simplifying assumptions adopted in the modeling process may limit prediction accuracy. As a result, with the rapid development of traffic data collection and storage technologies, research has increasingly shifted toward data-driven approaches. Lee et al. [25] compared fast-time simulation with common machine-learning methods and showed the advantages of the latter in terms of prediction errors. Since then, numerous machine-learning methods have been applied to taxi-time prediction and have demonstrated strong performance [3–6]. Recent studies have also explored deep learning methods such as graph neural networks to capture spatial interactions on the airport surface [26].

Although these studies show good performance, they often treat taxi-time prediction as an isolated task and do not examine whether the resulting predictions are suitable inputs to DM optimization. Feature selection is particularly important for this purpose. Dynamic features may improve prediction accuracy, but they are derived from pushback decisions, leading to a circular dependency between prediction and optimization. Prior analysis has shown that prediction accuracy does not necessarily require every available feature [2], indicating that suitable features should be selected before using the predictions in DM optimization. For example, current taxi-time-prediction-based DM studies select only features available before pushback to predict taxi time [1] or its distribution [11], which is a practical choice to avoid circular dependence. However, it remains valuable to investigate the improvements obtained by incorporating dynamic features compared with using only static features, and to analyze the trade-off between prediction accuracy and suitability for subsequent optimization. This research therefore develops both static-feature-based and dynamic-feature-enhanced prediction models and compares their performance. The results can provide insights into how to design prediction models that are not only accurate but also useful for decision support in airport surface operations.

3. Methodology

This section presents the proposed learning-based two-stage stochastic programming (SP) framework for departure metering (DM). As shown in Figure 2, the framework consists of three main components: prediction model training, distribution construction, and two-stage SP. First, a machine-learning (ML) model is trained offline to predict the taxi time of each departure flight based on historical operational data. For an operational instance of DM, the trained model is used to predict taxi times and their distributions based on current flight features. Based on the predicted taxi-time distributions, a two-stage SP model is developed. In the first stage, pushback times are determined before taxi times are realized, while in the second stage, runway takeoff times and sequences are adjusted to the realized taxiing conditions, and the resulting runway-area queueing cost serves as recourse for the first-stage DM decisions.

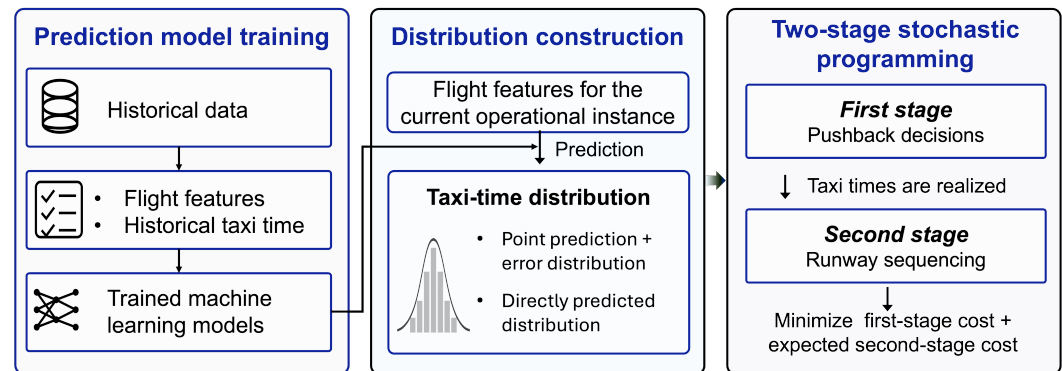


Figure 2. Learning-based two-stage stochastic programming framework for departure metering.

The proposed framework provides a general workflow for integrating ML-based taxi-time prediction with SP-based departure metering and can be applied to different airport configurations. It links ML-based prediction and SP-based optimization to capture flight-level taxi-time uncertainty and make more robust DM decisions. The following subsections provide the details of the components in the framework.

3.1. Prediction Model Training

This subsection describes the training of the ML models used for point and probabilistic taxi-time prediction. Following established practices in the literature, this research adopts similar design principles for models, features, and targets. The flight features used for prediction are summarized in Table 1, which can be extracted from the historical surface radar operational data of an airport. Note that the features used in this study are all static features, which can be observed before pushback. Although dynamic features describing surface traffic conditions may provide additional information for prediction, they depend on the pushback time, which is the decision variable of the optimization model. Incorporating such features would introduce a circular dependency between prediction and optimization, making the proposed sequential framework difficult to implement. Therefore, only static features are adopted in the proposed framework. The impact of excluding dynamic features on prediction performance is evaluated in the numerical experiments.

Table 1. Features for taxi-time prediction.

Name	Description
Terminal	Terminal of departure flight i .
Wake	Wake turbulence category of departure flight i .
Spot	Spot latitude and longitude of departure flight i .
Runway	Runway used for departure flight i .
Distance	Shortest path length from the spot to runway entry corresponding to flight i .
Weather	Temperature, northward/southward wind speed, and whether rainfall occurs when flight i pushes back.

The taxi time is defined as the time period from pushback to the runway-area entry threshold for each departure flight, similar to that in [1]. The runway-area entry threshold is defined as the position 200 m upstream of the runway entrance point along the taxi route. This definition, also called unimpeded taxi time, excludes the queueing time near the runway from taxi time, thereby reducing the impact of runway queueing time on taxi-time prediction.

For each flight sample in the historical data, let ξ_i denote the vector of features for flight i summarized in Table 1, and let T'_i denote its observed taxi time. Two types of machine-learning models are trained for taxi-time prediction: point prediction and distribution

prediction. Both types of models are trained using the observed samples (ξ_i, T_i') . The point-prediction model estimates a single taxi-time value based on the flight features ξ_i , whereas the distribution-prediction model estimates the probabilities associated with the possible taxi-time values based on ξ_i .

Several widely used machine-learning models are trained for both point prediction and distribution prediction, including K-Nearest Neighbors, Decision Tree, Random Forest, Extra Trees, Gradient Boosting, Histogram Gradient Boosting, AdaBoost, Multilayer Perceptron, XGBoost, and CatBoost. After training, the point-prediction model can be represented as

$$\hat{T}_i = f(\xi_i), \quad (1)$$

where $\hat{T}_i$ denotes the predicted taxi time of flight i . Similarly, the distribution-prediction model can be represented as

$$\hat{\mathbb{P}}_i = g(\xi_i), \quad (2)$$

where $\hat{\mathbb{P}}_i$ denotes the predicted conditional probability distribution of taxi time.

The functions f and g are the trained prediction models, and they are used to generate the taxi-time prediction and its distribution for each flight in the optimization model.

3.2. Distribution Construction

Using the trained prediction models, the taxi-time distribution for each flight is constructed from its features ξ_i according to the two approaches described below.

The first approach constructs the error-based distribution, which combines point predictions with empirical error distributions. The second approach directly uses the conditional distribution produced by the probabilistic prediction model.

For the error-based distribution, the taxi time of flight i is represented as

$$T_i = \hat{T}_i(1 + \varepsilon_i), \quad \forall i \in \mathcal{F}^{\text{dep}}, \quad (3)$$

where $\hat{T}_i = f(\xi_i)$ is the predicted taxi time of flight i from the point prediction model and ε_i is a random parameter. Here, ε follows an empirical distribution derived from the prediction errors in the test set. Therefore, the taxi-time distribution of flight i is constructed as

$$\hat{\mathbb{P}}_i^{\text{err}} : \hat{P}(T_i = \hat{T}_i(1 + \varepsilon_{i,\psi})) = P(\varepsilon_i = \varepsilon_{i,\psi}), \quad \forall \psi \in \Psi_i, \quad (4)$$

where Ψ_i is the set of possible taxi-time values for flight i and $\varepsilon_{i,\psi}$ is the ψ -th possible prediction error value. The empirical distribution of ε_i is constructed from the prediction errors of the test set, which is the difference between the predicted taxi time and the observed taxi time.

For the predicted distribution, each flight i is assigned a discrete conditional distribution $\hat{\mathbb{P}}_i^{\text{pred}} = g(\xi_i)$ directly provided by the probabilistic prediction model:

$$\hat{\mathbb{P}}_i^{\text{pred}} : \hat{P}(T_i = \tau_{i,k} | \xi_i), \quad \forall k \in \mathcal{K}_i, \quad (5)$$

where $\mathcal{K}_i$ is the set of possible taxi-time values for flight i and $\tau_{i,k}$ is the k -th possible taxi-time value.

3.3. Two-Stage Optimization Model

Using the constructed probability distribution for taxi times, a two-stage stochastic programming (SP) model is formulated to optimize departure pushback times under taxi-time uncertainty.

3.3.1. Model Formulation

The notation used in the model is summarized in Table 2, where arrival–arrival ordering is fixed by the scheduled runway-exit times and is therefore not represented by an optimization variable.

Table 2. Notation used in the SP model.

Symbol	Description
Sets	
$\mathcal{F}^{\text{dep}}$	Set of departure flights.
$\mathcal{F}^{\text{arr}}$	Set of arrival flights.
$\mathcal{R}_i^{\text{dep}}$	Set of departure flights, excluding flight i when applicable, that have the same runway as flight i , $\forall i \in \mathcal{F}^{\text{dep}} \cup \mathcal{F}^{\text{arr}}$.
$\mathcal{R}_i^{\text{arr}}$	Set of arrival flights, excluding flight i when applicable, that have the same runway as flight i , $\forall i \in \mathcal{F}^{\text{dep}} \cup \mathcal{F}^{\text{arr}}$.
$\mathcal{I}^{\text{DD}}$	Set of unordered departure–departure flight pairs sharing the same runway, defined as $\{(i, j) : i \in \mathcal{F}^{\text{dep}}, j \in \mathcal{R}_i^{\text{dep}}, i \prec j\}$ under a fixed flight index order $\prec$.
$\mathcal{I}^{\text{DA}}$	Set of departure–arrival flight pairs sharing the same runway, defined as $\{(i, k) : i \in \mathcal{F}^{\text{dep}}, k \in \mathcal{R}_i^{\text{arr}}\}$.
Parameters	
s_i	Scheduled pushback time of departure flight i , $\forall i \in \mathcal{F}^{\text{dep}}$.
a_i	Scheduled runway-exit time for arrival flight i , $\forall i \in \mathcal{F}^{\text{arr}}$.
$\bar{d}$	Maximum allowable pushback schedule deviation for each departure flight.
q	Unimpeded queueing time of flight i from the runway threshold to the runway entrance position.
$\bar{q}$	Maximum queueing time of flight i at the runway area.
$\Delta_{i,j}$	Minimum separation time between successive departure flights i and j , determined according to the wake categories of the aircraft.
u_k	Occupation time of flight k at the runway area, $\forall k \in \mathcal{F}^{\text{arr}}$.
v_k	Buffer time after the occupation of flight k at the runway area, $\forall k \in \mathcal{F}^{\text{arr}}$.
T_i	Random parameter representing the taxi time of flight i , $\forall i \in \mathcal{F}^{\text{dep}}$.
Decision variables	
x_i	Pushback time of flight i , $\forall i \in \mathcal{F}^{\text{dep}}$.
e_i	Entry runway area time of flight i , $\forall i \in \mathcal{F}^{\text{dep}}$.
y_i	Departure time of flight i , $\forall i \in \mathcal{F}^{\text{dep}}$.
q_i	Queueing time of flight i at the runway area, $\forall i \in \mathcal{F}^{\text{dep}}$.
$\eta_{i,j}$	Binary variable equal to 1 if departure flight i is ordered before departure flight j , and 0 otherwise, $\forall (i, j) \in \mathcal{I}^{\text{DD}}$.
$\gamma_{i,k}$	Binary variable equal to 1 if departure flight i is ordered before arrival flight k , and 0 otherwise, $\forall (i, k) \in \mathcal{I}^{\text{DA}}$.

The SP model is formulated as a two-stage model. The first-stage decision is to determine the pushback time of each departure flight, and the second-stage decision is to determine the departure time of each departure flight after the realization of taxi time. The objective function consists of two parts: the first part is the penalty cost for schedule deviation, and the second part is the expected total queueing time at the runway area.

$$\min_x \left\{ \lambda \cdot \sum_{i \in \mathcal{F}^{\text{dep}}} (x_i - s_i) + \mathbb{E}_{\mathbb{P}}[Q(\mathbf{x}, T)] \right\}, \quad (6)$$

$$\text{s.t. } x_i \geq s_i, \forall i \in \mathcal{F}^{\text{dep}}, \quad (7)$$

$$x_i - s_i \leq \bar{d}, \forall i \in \mathcal{F}^{\text{dep}}. \quad (8)$$

The first-stage problem minimizes the total cost of pushback schedule deviation (weighted by λ) and expected queueing time $\mathbb{E}_{\mathbb{P}}[Q(\mathbf{x}, T)]$ at the runway area by optimizing

the pushback time of each departure flight. Each flight is constrained to push back no earlier than its scheduled time in Constraint (7) and no later than the latest allowable time window in Constraint (8). $Q(x, T)$ is the optimal value of the second-stage problem given the first-stage decision x and the realization of taxi time T .

The second-stage problem is formulated as follows:

$$Q(x, T) = \min_{e, y, q, \eta, \gamma} \sum_{i \in \mathcal{F}^{\text{dep}}} q_i, \quad (9)$$

$$\text{s.t. } e_i = x_i + T_i, \forall i \in \mathcal{F}^{\text{dep}}, \quad (10)$$

$$y_i = e_i + q_i, \forall i \in \mathcal{F}^{\text{dep}}, \quad (11)$$

$$q_i \geq \underline{q}, \forall i \in \mathcal{F}^{\text{dep}}, \quad (12)$$

$$q_i \leq \bar{q}, \forall i \in \mathcal{F}^{\text{dep}}, \quad (13)$$

$$e_i \leq e_j \Rightarrow \eta_{i,j} = 1, \forall (i, j) \in \mathcal{I}^{\text{DD}}, \quad (14)$$

$$e_j \leq e_i \Rightarrow \eta_{i,j} = 0, \forall (i, j) \in \mathcal{I}^{\text{DD}}, \quad (15)$$

$$\eta_{i,j} = 1 \Rightarrow y_j \geq y_i + \Delta_{i,j}, \forall (i, j) \in \mathcal{I}^{\text{DD}}, \quad (16)$$

$$\eta_{i,j} = 0 \Rightarrow y_i \geq y_j + \Delta_{j,i}, \forall (i, j) \in \mathcal{I}^{\text{DD}}, \quad (17)$$

$$\gamma_{i,k} = 1 \Rightarrow y_i \leq a_k - u_k, \forall (i, k) \in \mathcal{I}^{\text{DA}}, \quad (18)$$

$$\gamma_{i,k} = 0 \Rightarrow y_i \geq a_k + v_k, \forall (i, k) \in \mathcal{I}^{\text{DA}}, \quad (19)$$

$$\eta_{i,j} \in \{0, 1\}, \forall (i, j) \in \mathcal{I}^{\text{DD}}, \quad (20)$$

$$\gamma_{i,k} \in \{0, 1\}, \forall (i, k) \in \mathcal{I}^{\text{DA}}. \quad (21)$$

The second-stage problem minimizes the total queueing time at the runway area given the first-stage pushback decisions and the realization of taxi time by optimizing the departure order and departure time of each departure flight. Equation (9) defines the objective function of the second stage problem, which minimizes the total queueing time. Since the second-stage problem is based on first-stage decisions x and the realization of taxi time T , the optimal value of the objective is denoted as $Q(x, T)$. Constraint (10) defines the relationship between entry runway area time and pushback time, and constraint (11) defines the relationship between departure time, entry runway area time, and queueing time. Constraints (12) and (13) set the lower and upper bounds of queueing time, respectively. The lower bound corresponds to the unimpeded queueing time, indicating the minimum time required for a flight to taxi from the runway threshold to the runway entrance position when there is no queue. The upper bound is set manually to prevent excessively long queueing time for a subset of flights. To some extent, these constraints prevent individual flights from experiencing excessively long queueing times when the model prioritizes overall system efficiency. Constraints (16) and (17) ensure that the wake-category-dependent minimum separation time is maintained between any two departure flights that take off successively. The separation parameter $\Delta_{i,j}$ is determined according to the wake categories of the leading flight i and trailing flight j with the corresponding separation values summarized in Table A2. These separation constraints are defined under normal runway operating conditions. Therefore, temporary operational changes caused by adverse weather conditions (e.g., deicing operations) are not considered in this study. Constraints (14) and (15) describe the first-come-first-served (FCFS) policy for departure runway usage. When FCFS is not enforced, these constraints could be deleted. Constraints (18) and (19) ensure that the departure time of each departure flight does not conflict with the occupation time of any arrival flight at the runway area.

This formulation uses one binary variable for each unordered departure–departure pair and one binary variable for each departure–arrival pair. It is equivalent to the bidirec-

tional $\sigma_{i,j}$ formulation, but it avoids creating fixed arrival–arrival variables and removes the redundant paired variables and mutual-exclusion constraints. If no departure–arrival pair shares a runway in the selected time window, then $\mathcal{I}^{\text{DA}}$ is empty and no arrival-related binary variables or constraints are generated.

Note that constraints (14)–(19) contain logical indicator constraints. They can be linearized by introducing sufficiently large constants M as follows:

$$M \cdot \eta_{i,j} \geq e_j - e_i, \forall (i, j) \in \mathcal{I}^{\text{DD}}, \quad (22)$$

$$M \cdot (1 - \eta_{i,j}) \geq e_i - e_j, \forall (i, j) \in \mathcal{I}^{\text{DD}}, \quad (23)$$

$$y_j \geq y_i + \Delta_{i,j} - M \cdot (1 - \eta_{i,j}), \forall (i, j) \in \mathcal{I}^{\text{DD}}, \quad (24)$$

$$y_i \geq y_j + \Delta_{j,i} - M \cdot \eta_{i,j}, \forall (i, j) \in \mathcal{I}^{\text{DD}}, \quad (25)$$

$$y_i \leq a_k - u_k + M \cdot (1 - \gamma_{i,k}), \forall (i, k) \in \mathcal{I}^{\text{DA}}, \quad (26)$$

$$y_i \geq a_k + v_k - M \cdot \gamma_{i,k}, \forall (i, k) \in \mathcal{I}^{\text{DA}}. \quad (27)$$

Using $\mathcal{X}$ and $\mathcal{Y}(x, T)$ to represent the feasible region of the first-stage decision and the second-stage decision, respectively, the matrix-vector form of the SP model is formulated as follows:

$$\text{SP-EXP: } \min_{x \in \mathcal{X}} \{ \lambda \cdot \|x - s\|_1 + \mathbb{E}_{\mathbb{P}}[Q(x, T)] \}. \quad (28)$$

3.3.2. Model Approximation Reformulation

This subsection introduces the solution approach to the two-stage SP problem defined in (28). It consists of two main components: model approximation reformulation to make the expectation term tractable and an implementation setting that reflects the local decision scope of practical DM.

The reformulation consists of two main steps: first, sample average approximation (SAA) is used to reformulate the expectation of queueing cost as a finite-sample average [27]; second, a scenario generation (SG) procedure is applied to further reduce the number of scenarios and make the optimization problem tractable [28]. The resulting SP-SG model is then solved for the flights that belong to the current DM decision scope.

Using SAA, the joint distribution of taxi time is approximated by a finite set of N samples:

$$\{\tilde{T}^1, \dots, \tilde{T}^N\}. \quad (29)$$

These samples have the same probability of occurrence, i.e., $P(T = \tilde{T}^n) = \frac{1}{N}$, $\forall n = 1, \dots, N$. They are sampled from the distributions constructed in Section 3.2. Using these samples to approximate the expectation term in the objective function, the reformulated model is as follows:

$$\text{SP-SAA: } \min_{x \in \mathcal{X}} \left\{ \lambda \cdot \|x - s\|_1 + \frac{1}{N} \sum_{n=1}^N Q(x, \tilde{T}^n) \right\}. \quad (30)$$

Note that when the number of samples N is large, directly solving the SAA model may become computationally expensive. Therefore, an SG procedure is further applied to compress the original sample set into a smaller representative scenario set. The SG procedure is achieved by solving the p-median-type model in Appendix A, which selects representative taxi-time vectors and assigns each original sample to one selected scenario.

By using SG, a reduced scenario set $\{\tilde{T}^l\}_{l=1}^L$ is obtained with associated probabilities $\{P^{(l)}\}_{l=1}^L$, where $L \ll N$ [28]. The SG-based SP model is then formulated as follows:

$$\text{SP-SG: } \min_{x \in \mathcal{X}} \left\{ \lambda \cdot \|x - s\|_1 + \sum_{l=1}^L P^{(l)} \cdot Q(x, \tilde{T}^l) \right\}. \quad (31)$$

By adjusting the number of scenarios L , the trade-off between solution quality and computational efficiency can be balanced. Specifically, L can be reduced to ensure tractability within manageable computational time. On the other hand, L can also be increased to obtain a more comprehensive representation of the uncertainty, thereby improving the first-stage solution quality. The preference depends on the scale of the problem and priorities in real-world applications.

3.3.3. Rolling Horizon Implementation

Note that in real-world operations, the scheduled times for departures s_i and arrivals a_i are usually determined shortly before the flights are ready to pushback or arrive and updated over time. So DM usually only makes decisions for the flights that are ready to pushback or will be ready to pushback in the near future. For departures that have already pushed back or whose pushback times are already determined, their pushback times are fixed and cannot be adjusted. For departures that are scheduled to pushback in the far future, their scheduled pushback times are not completely determined yet and it's hard to optimize their pushback times based on the current information. Thus, in this research, at the current time step τ , the DM-SP-SG only optimizes the flights scheduled to pushback in a near-future period with length H^O . Departures scheduled to pushback in the recent past period with length H^B and departures shortly after the optimized period with length H^A are also considered in the model, but their pushback times are fixed.

Mathematically, according to the latest updated s , the set of departures considered by DM is

$$\mathcal{W}_\tau = \left\{ i \in \mathcal{F}^{\text{dep}} : \tau - H^B \leq s_i < \tau + H^O + H^A \right\}. \quad (32)$$

This set is divided into three parts:

$$\mathcal{B}_\tau = \left\{ i \in \mathcal{F}^{\text{dep}} : \tau - H^B \leq s_i < \tau \right\}, \quad (33)$$

$$\mathcal{O}_\tau = \left\{ i \in \mathcal{F}^{\text{dep}} : \tau \leq s_i < \tau + H^O \right\}, \quad (34)$$

$$\mathcal{A}_\tau = \left\{ i \in \mathcal{F}^{\text{dep}} : \tau + H^O \leq s_i < \tau + H^O + H^A \right\}. \quad (35)$$

Only flights in $\mathcal{O}_\tau$ are optimized by the current DM optimization. Pushback times for flights in $\mathcal{B}_\tau$ have already been determined, so their pushback times are fixed to the decisions made at previous time steps $\tau - 1$. Flights in $\mathcal{A}_\tau$ are outside the current optimized period, so their pushback times are fixed at their latest updated scheduled values. These fixed flights are still considered in the second-stage runway sequencing and separation constraints, so their effects on runway queueing are kept even though their pushback times are not adjustable decision variables.

The DM-SP-SG problem solved at the current time τ is formulated as

$$\text{DM-SP-SG}_\tau : \min_{x^\tau} \left\{ \lambda \sum_{i \in \mathcal{O}_\tau} |x_i^\tau - s_i^\tau| + \sum_{l=1}^L P_\tau^{(l)} Q_\tau(x^\tau, \tilde{T}_\tau^l) \right\}, \quad (36)$$

$$\text{s.t. } x_i^\tau \geq s_i, \quad \forall i \in \mathcal{O}_\tau, \quad (37)$$

$$x_i^\tau \leq s_i + d, \quad \forall i \in \mathcal{O}_\tau, \quad (38)$$

$$x_i^\tau = x_i^{\tau-1,*}, \quad \forall i \in \mathcal{B}_\tau, \quad (39)$$

$$x_i^\tau = s_i, \quad \forall i \in \mathcal{A}_\tau. \quad (40)$$

For real-time implementation, at each time step τ , s_i is updated for each flight i based on the latest information. The flights within the current DM decision scope are then identified, and the DM-SP-SG problem is solved to obtain the optimized pushback times $x_i^{\tau,*}$ for flights in $\mathcal{O}_\tau$. Once the information is updated at the next time step $\tau + 1$, these steps are repeated until the end of the day.

4. Numerical Experiment

In this section, the proposed learning-based two-stage SP framework is applied to a numerical experiment based on real-world data from Tokyo International Airport (RJTT). The prediction models are trained as described in Section 3.1 using data in 2021 and 2022. The pushback time on 8 May 2023 is optimized, which is a typical north-wind-operation day as defined in Kato and Itoh [1]. (Other operation days are also tested and the results are summarized in Appendix D). The predicted distribution and error-based distribution are constructed based on the approaches described in Section 3.2. The actual pushback times in operations are regarded as scheduled pushback times. For SP implementation, the SAA method selects $N = 1000$ equally likely samples from the constructed distributions and SG then generates $L = 100$ typical scenarios with corresponding probabilities. To emulate the real-time implementation of DM, the DM optimization formulated in Equations (36)–(40) is executed every 10-min rolling interval. At each execution time, the model optimizes departures in the next 20 min, while departures that pushed back within the previous 20 min and departures planned between 20 and 40 min after the current time are also included with fixed pushback times to preserve their boundary effects. Here, the rolling interval is set to shorter than the length of optimization horizon H^O , so some flights optimized in time step τ can be adjusted again in the next time step $\tau + 1$. The main optimization parameter settings are set based on historical data or operational rules and summarized in Appendix C. Each optimization model is solved using Gurobi 12.0 with a time limit of 5 min and default settings for other solver parameters [29]. According to the preliminary experiments, this setting can achieve a good balance between solution quality and computational efficiency. Two different distributions (predicted and error-based) are used and the models are implemented under both sequencing (without Constraints (14) and (15)) and FCFS (with Constraints (14) and (15)) second-stage settings. The results are analyzed as follows.

4.1. Prediction Results and Constructed Distribution

The surface-radar data at Tokyo International Airport (RJTT) are used to train and evaluate the prediction models. Data from 2021 and 2022 are used for training, and data in 2023 are reserved for testing. Recall that the prediction models in Section 3.1 use only the static features in Table 1. To test the impact of dynamic features that indicate the surface traffic conditions but rely on the pushback-time decisions, a dynamic-feature-enhanced specification is further considered in this experiment. Table 3 lists the additional dynamic features that are commonly used in existing literature. They describe the contemporaneous surface traffic state but depend on the pushback time.

Some commonly used metrics are introduced to evaluate the prediction performance of the models. For point prediction, the metrics include mean absolute error (MAE), root mean squared error (RMSE), coefficient of determination (R^2), and mean absolute percentage error (MAPE). Lower MAE, RMSE, and MAPE values and a higher R^2 value are preferred. For distribution prediction, the metrics are LogLoss, continuous ranked probability score

(CRPS), prediction interval coverage probability (PICP), and mean prediction interval width (MPIW). The definitions of these metrics are provided in Appendix B. Lower LogLoss and CRPS are preferred, while PICP should be close to its nominal 90% level without an excessively large MPIW.

Table 3. Additional dynamic features.

Name	Description
H_{pb}, M_{pb}	Hour and minute of the pushback time of flight i .
$A^{e/l}$	Number of departures pushing back within the five minutes before/after flight i .
$D^{e/l}$	Number of arrivals starting to taxi within the five minutes before/after flight i .
$A_{tml/rwy}^{e/l}$	Number of departures associated with the same terminal or runway and pushing back within the five minutes before/after flight i .
$D_{tml/rwy}^{e/l}$	Number of arrivals associated with the same terminal or runway and starting to taxi within the five minutes before/after flight i .
N_{share}	Number of flights sharing taxi-route segments with flight i .
$N_{conflict}$	Number of flights with potential taxi-route conflicts with flight i , estimated from taxi routes and historical taxi speeds.

Table 4 and Table 5 report the four best-performing point-prediction and distribution-prediction models, respectively. Each value outside parentheses is the test-set performance of the static-feature model, whereas the value in parentheses is the change obtained after adding the dynamic features.

Table 4. Taxi-time prediction performance.

Model	MAE	RMSE	R^2	MAPE
CatBoost	92.68 (−2.11)	129.55 (−3.42)	0.43 (+0.03)	12.25% (−0.20%)
HistGB	93.46 (−2.57)	130.03 (−3.61)	0.43 (+0.03)	12.40% (−0.30%)
XGBoost	93.34 (−2.48)	130.08 (−3.29)	0.43 (+0.03)	12.37% (−0.29%)
GBoost	94.40 (−2.15)	131.55 (−3.12)	0.42 (+0.03)	12.49% (−0.24%)

Table 5. Distribution prediction performance.

Model	LogLoss	CRPS (s)	PICP	MPIW (s)
CatBoost	3.93 (−0.04)	74.59 (−1.93)	86.89% (+2.97%)	377.06 (+45.03)
RandomForest	4.09 (+0.36)	73.64 (−1.92)	87.45% (+3.65%)	382.28 (+62.22)
ExtraTrees	4.11 (+0.14)	73.34 (−2.13)	87.61% (+2.98%)	386.60 (+39.62)
XGBoost	4.12 (−0.03)	75.25 (−2.15)	84.09% (+1.60%)	343.18 (+27.42)

Considering the point- and distribution-prediction results together, CatBoost provides the best overall performance. It is the best point-prediction model and achieves the lowest LogLoss for distribution prediction while remaining competitive in CRPS and interval coverage. Therefore the CatBoost model is used for all subsequent point predictions and predicted distributions.

The two experiments also show that dynamic features provide only marginal improvements. They reduce point-prediction MAE by 2.11–2.57 s and distributional CRPS by about 2 s, while their effect on LogLoss is inconsistent. Although PICP increases, the corresponding prediction intervals become 27–62 s wider. These results demonstrate that adding these dynamic features yields little overall improvement.

The limited improvement from dynamic features may have two explanations. First, although taxi time is affected by surface traffic conditions, the current dynamic features

provide only a coarse representation and cannot capture the complex spatiotemporal interactions between aircraft on the surface. Second, the taxiing process is influenced by interventions from ground controllers, who may adjust aircraft taxiing behavior to mitigate congestion and conflicts. This human intervention may weaken the correlation between surface traffic conditions and taxi time, making it difficult for the model to learn from dynamic features. Thus, considering the limited improvement provided by the current dynamic features and the fact that they introduce a circular dependency with the pushback-time decision, it is reasonable to use only static features for prediction.

Using the trained prediction model, the distribution of taxi time for each flight is constructed. Table 6 evaluates the two distribution-construction approaches described in Section 3.2. For the error-based variants, errors in the test set are either pooled across all flights or grouped according to the point prediction value using intervals of 100, 200, or 500 s. All variants are evaluated on the same test samples and discrete taxi-time support.

Table 6. Predicted distribution and error-based distribution comparison.

Model	LogLoss	CRPS (s)	PICP	MPIW (s)
Predicted	3.93	74.59	86.89%	377.06
All-error-based	5.27	74.90	79.98%	296.16
100 s-error-based	6.19	75.38	80.22%	298.36
200 s-error-based	5.97	75.20	80.16%	298.36
500 s-error-based	5.31	74.90	79.96%	295.22

The directly predicted distribution performs best overall: it has substantially lower LogLoss, comparable or better CRPS, and coverage much closer to the nominal 90% level than all error-based variants. Its wider interval reflects more reliable uncertainty coverage rather than unnecessary dispersion, because the narrower error-based intervals miss the observed taxi time more often. This confirms that flight-conditioned distributions capture heterogeneous uncertainty more effectively than shared empirical error distributions. Therefore, the directly predicted distribution is used as the primary uncertainty representation and the 100-s error grouping, which has the highest PICP among the empirical variants, is retained as the baseline.

4.2. Computational Details

The convergence performance is first evaluated, measured by the Gurobi gaps across the repeated DM executions. The Gurobi gap is defined as the relative difference between the incumbent objective value and the best objective bound according to Gurobi Optimization, LLC [29]:

$$\text{Gurobi gap} = \frac{|z_{\text{inc}} - z_{\text{bound}}|}{|z_{\text{inc}}|}, \quad (41)$$

where z_{inc} denotes the incumbent objective value and z_{bound} denotes the best objective bound. Figure 3 shows the temporal distribution of the Gurobi gap of each solved DM-SP-SG instance, where the marker color represents the number of departures included in the corresponding problems. Overall, acceptable solutions are achieved in most instances, where the convergence is easier under FCFS settings. In the sequencing settings, the median gaps are 3.5% for the predicted distribution and 3.1% for the error-based distribution, and 61.3% and 64.2% of the instances, respectively, finish within a 5% gap. In contrast, the FCFS cases show stronger convergence, with both median gaps equal to 0 and 86.8% of FCFS instances with predicted distribution and 84.9% of FCFS instances with error-based distribution finishing within a 5% gap. This pattern is expected because the FCFS constraints fix the takeoff order in the second stage, whereas the sequencing model must

search over a larger set of feasible runway sequences, making the two-stage optimization more challenging.

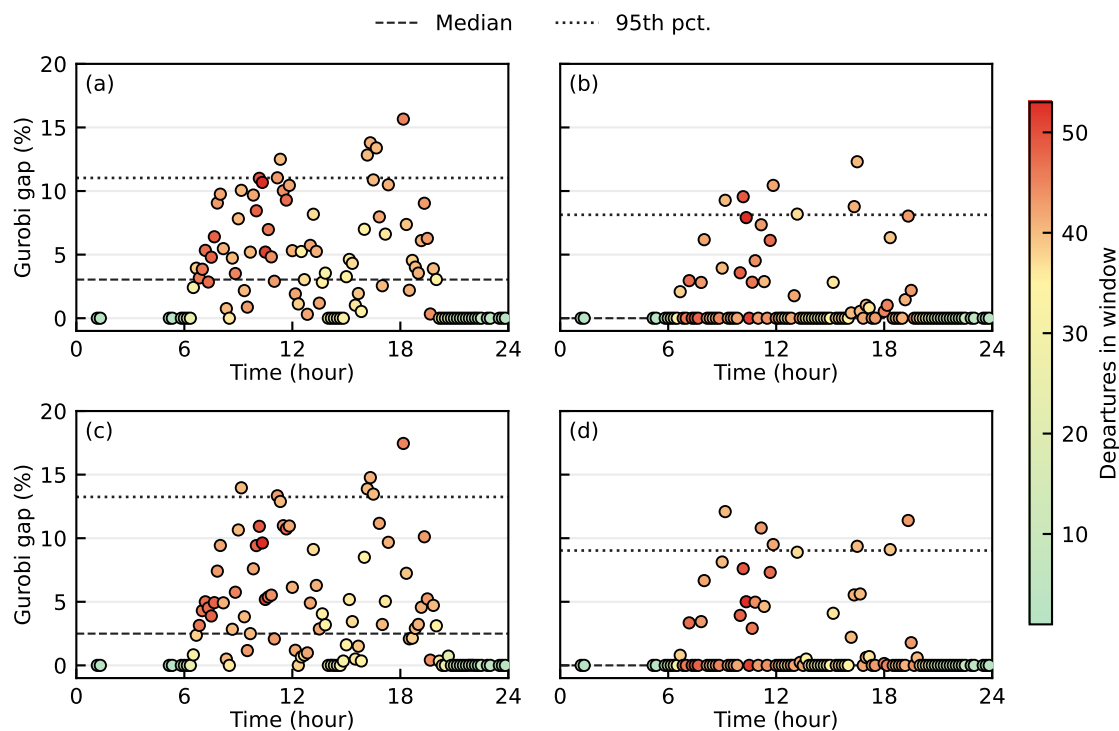


Figure 3. Distribution of Gurobi gaps across repeated DM executions: (a) predicted taxi times with optimized sequencing; (b) predicted taxi times with FCFS; (c) error-based taxi times with optimized sequencing; and (d) error-based taxi times with FCFS.

Furthermore, DM instances that are difficult to converge are mainly associated with peak periods with a large number of flights. The color scale in Figure 3 shows that many high-gap points appear in warmer colors, indicating larger departures in the corresponding DM window. This is consistent with the general intuition that denser departure traffic leads to larger and more difficult optimization problems. However, there are also exceptions showing that the relationship is not purely monotonic. For example, compared to the 10:00–12:00 period, the 17:30–18:00 period contains fewer departures, but usually corresponds to larger gaps. It is found that the 17:30–18:00 instances combine a large number of departures with a large number of arrivals and strong boundary effects from fixed flights before and after the optimized window. In particular, the 17:30–18:00 windows contain about 42 departures and 42–43 arrivals, with 17–18 optimized departures and many fixed neighboring departures. This suggests that departure count, although directly determining the problem size, captures only one dimension of the problem complexity. Overall, the main reason leading to convergence difficulties is a combination of traffic density and interaction complexity rather than departure count alone.

4.3. Pushback Displacement

The temporal distribution of pushback displacement is next analyzed. Figure 4 summarizes the total displacement and average displacement per flight in each hour. It can be seen that there are two peaks of total displacement, which correspond to the late morning peak around 8:00–13:00 and the afternoon peak around 16:00–18:00. This distribution is consistent with that of the average displacement, which indicates that the peak periods are not only caused by higher traffic volume but also by stronger per-flight interventions,

implying tighter system constraints and more intensive congestion regulation. This also illustrates that the current pushback schedule urgently needs adjustment, especially for peak hours.

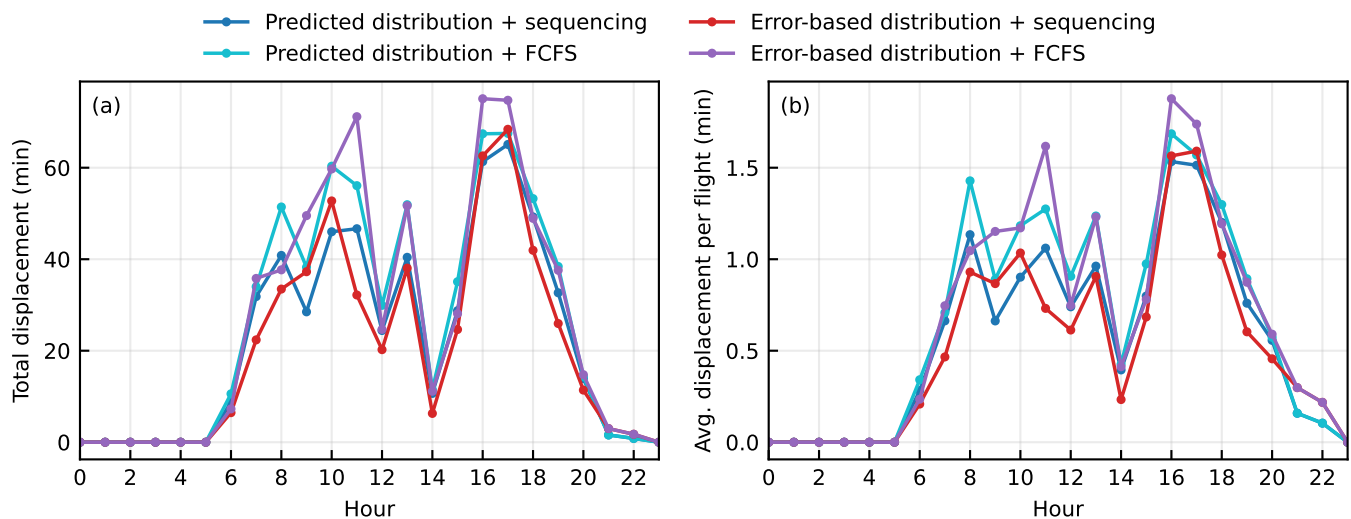


Figure 4. Hourly pushback displacement: (a) total displacement; and (b) average displacement per flight.

The displacement patterns under the four settings are broadly similar, but their magnitudes differ across time periods. For the same second-stage queueing policy, the error-based distribution generally leads to larger displacement than the predicted distribution in the congested periods, and this pattern is particularly clear under the FCFS setting. This indicates that the error-based construction produces a more conservative first-stage schedule. It is interesting because the error-based distributions show narrower ranges (smaller MPIWs) in Table 6 but incur more conservative decisions. This phenomenon suggests that narrower prediction intervals do not necessarily provide a more accurate representation of the underlying taxi-time distributions. For the same distribution construction method, the FCFS setting usually produces larger displacement than sequencing across most active hours, which is intuitive. This means that when runway order cannot be adjusted in the second stage, the model has to rely more on first-stage pushback displacement to prevent queueing conflicts. By contrast, flexible runway sequencing absorbs part of the queueing interaction, reducing runway queueing and the need for large pushback adjustments.

4.4. Solution Quality

The quality of the optimized pushback schedules is further evaluated using independently sampled Monte Carlo taxi-time scenarios. The first-stage pushback times are fixed during this evaluation. A total of 1000 taxi-time scenarios are sampled independently from the predicted flight-level probability distributions, and the second-stage problem (9)–(21) is solved for each scenario. The three tested periods, 10:00–12:00, 12:00–14:00, and 16:00–18:00, are used as multiple operating scenarios with different traffic levels. Here, all flights in the two-hour block are evaluated together in one second-stage problem, rather than by repeatedly executing DM over the two-hour period. In addition to the schedules optimized by SP models based on the predicted and the error-based distributions, a certainty model is included where the taxi-time is the point-prediction value $\hat{T}_i$. The distributions are shown in Figure 5, where the dotted line represents the mean of the distribution.

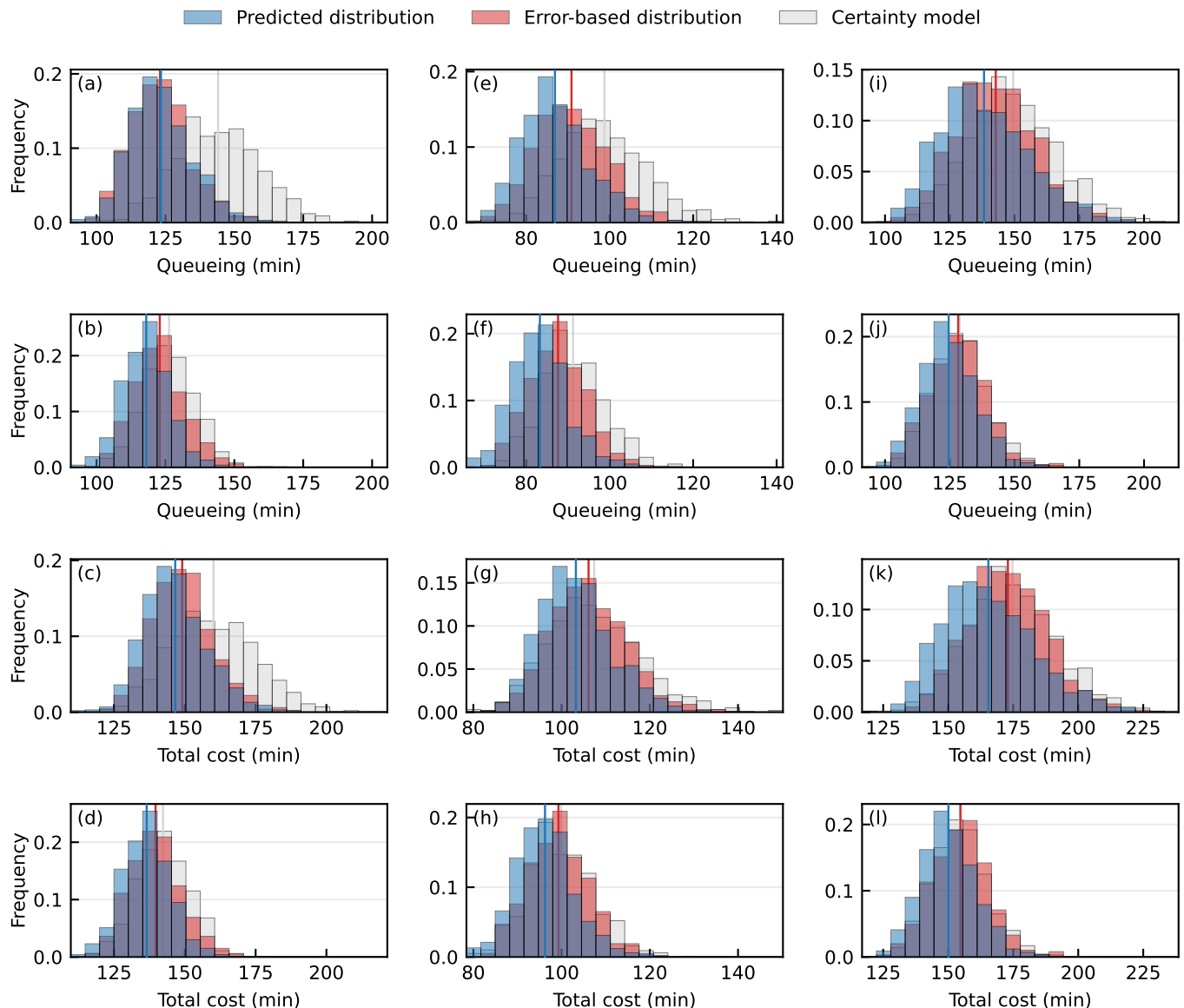


Figure 5. Distributions of second-stage queueing time and weighted total cost. (a) FCFS queueing time from 10:00–12:00; (b) sequencing queueing time from 10:00–12:00; (c) FCFS weighted total cost from 10:00–12:00; (d) sequencing weighted total cost from 10:00–12:00; (e) FCFS queueing time from 12:00–14:00; (f) sequencing queueing time from 12:00–14:00; (g) FCFS weighted total cost from 12:00–14:00; (h) sequencing weighted total cost from 12:00–14:00; (i) FCFS queueing time from 16:00–18:00; (j) sequencing queueing time from 16:00–18:00; (k) FCFS weighted total cost from 16:00–18:00; and (l) sequencing weighted total cost from 16:00–18:00. Blue, red, and gray vertical lines represent the mean queueing time or total cost of SP, error-based SP, and certainty model, respectively.

The analysis begins with queueing time distributions, shown in the first two rows of Figure 5. The certainty model consistently produces distributions that are shifted to the right, with a heavier right tail, indicating a higher probability of large queueing outcomes. In contrast, the predicted-distribution SP leads to a clear leftward shift and a more concentrated mass, showing that both the overall level and variability of queueing time are reduced. This pattern holds under both FCFS and sequencing. Specifically, the predicted-distribution SP reduces mean queueing time by 7.5–14.4% under FCFS and by 3.3–6.5% under sequencing. Compared with the certainty model, the error-based SP, regarded as an empirical baseline discussed in Section 3.2, also shifts the distribution leftward, but its improvement is more limited and the right tail remains heavier than that of the predicted-

distribution SP. In particular, under sequencing, the error-based SP distributions still exhibit noticeable overlap with the certainty model, whereas the predicted-distribution SP is more clearly separated. This suggests that the improvement depends not only on adopting an SP framework but also on how accurately taxi-time uncertainty is represented. The certainty model ignores the dispersion and asymmetry of uncertainty, while the error-based approach only provides a coarse approximation based on residuals.

The analysis then turns to the weighted-total-cost distributions shown in the third and fourth rows of Figure 5. The predicted-distribution SP produces distributions that are consistently shifted to lower values while remaining well concentrated, without exhibiting a noticeable increase in dispersion or heavier right tails. This indicates that reductions in queueing time are not accompanied by additional variability or extreme outcomes in total cost. Therefore, the lower queueing time is not achieved by simply adding excessive planned pushback holding; it translates into a lower overall objective value after accounting for first-stage displacement. These distributional patterns further suggest that the improvement of the proposed framework comes from a more effective coordination between the two stages, where pushback decisions better anticipate the realized taxi-time uncertainty.

4.5. Flight-Level Queueing Equity Analysis

Flight-level queueing equity is further examined to determine whether the queueing reductions achieved by the SP methods come at the expense of a small subset of flights. This experiment uses the same three operating periods and the same 1000 independently sampled taxi-time scenarios as in Section 4.4. Flight-level second-stage queueing times are evaluated under two SP settings, predicted-distribution SP with sequencing and predicted-distribution SP with FCFS, and compared with those under the certainty-model FCFS baseline. In each case, the queueing times in each test scenario are evaluated using the same sequencing or FCFS rule as in the corresponding optimization model. For each departure flight i , its second-stage queueing time q_i^ω is collected across all sampled scenarios $\omega \in \Omega$ under each setting, and two flight-level metrics are defined.

$$\bar{q}_i = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} q_i^\omega, \forall i \in F^{\text{dep}}, \quad (42)$$

$$Q_i^{90} = \text{Quantile}_{0.90}(\{q_i^\omega : \omega \in \Omega\}), \forall i \in F^{\text{dep}}. \quad (43)$$

Here, $\bar{q}_i$ measures the average queueing cost of flight i , while Q_i^{90} measures its tail queueing cost under extreme scenarios. These metrics characterize the distributions of average and tail queueing costs across flights to evaluate flight-level equity. Moreover, the Gini coefficient is introduced to quantify system inequality more clearly and is defined as

$$G(z) = \frac{\sum_{i \in F} \sum_{j \in F} |z_i - z_j|}{2|F|^2 \bar{z}}, \quad (44)$$

$$\bar{z} = \frac{1}{|F|} \sum_{i \in F} z_i. \quad (45)$$

Here, z_i can be either $\bar{q}_i$ or Q_i^{90} . According to the definition, $G(z)$ varies between 0 and 1, where a value of 0 indicates perfect equality (all flights have the same z_i), and a value close to 1 indicates high inequality (a small number of flights have much higher z_i than the rest). The Gini coefficient provides a summary measure of the dispersion of queueing costs across flights.

Figure 6 shows the distributions of $\bar{q}_i$ and Q_i^{90} across flights, and the calculated $G(\bar{q}_i)$ and $G(Q_i^{90})$ are labeled in the upper right corner of each subfigure. Overall, the two SP settings do not produce extreme distributions of second-stage queueing time across

flights. Specifically, the mean queueing time $\bar{q}_i$ is mostly concentrated between 1–2 min, meaning that the mean queueing time is relatively evenly distributed across flights. When it comes to the tail queueing time Q_i^{90} , although the distributions are more dispersed, the patterns for the SP and certainty settings are broadly similar. This similarity suggests that the dispersed distribution pattern for the tail queueing time is mainly an inherent characteristic of the problem, and the proposed SP methods do not introduce additional unevenness in queueing across flights compared with the certainty model. The calculated $G(\bar{q}_i)$ and $G(Q_i^{90})$ further support this observation, with values mostly between 0.16 and 0.31, indicating a moderate level of equity. In addition, the Gini coefficient values for SP are generally at the same level as or even lower than those for the certainty model, also suggesting that the SP methods usually do not undermine equity in queueing across flights. Overall, the queueing reduction of the proposed framework is not achieved at the expense of a small subset of flights, but rather through a more effective coordination between pushback decisions and runway sequencing under uncertainty. Such an equitable distribution of queueing outcomes may also improve stakeholder acceptance of DM interventions in practical airport operations.

4.6. Sensitivity Analysis

A local sensitivity analysis is further conducted to examine how the optimized schedule reacts when the scheduled pushback times s_i and arrival times a_i change after the DM decision. This analysis focuses on a partial problem in which the scheduled times s_i and a_i are perturbed after the initial DM decisions have been made. Specifically, assuming that the current time τ is 10:00, the pushback times scheduled in the next 20 min (10:00–10:20) are optimized using predicted-distribution SP with FCFS, while for flights scheduled in the preceding 20 min (9:40–10:00) and the following 20 min (10:20–10:40), their pushback times are fixed according to Equations (36)–(40). The optimized pushback times are recorded as x_i^* for each flight i . s_i and a_i of flights scheduled in the future 10:00–10:40 are perturbed. The perturbation is generated independently as

$$s'_i = s_i + \delta_i, \quad \delta_i \sim \mathcal{N}\left(0, (\alpha(s_i - \tau))^2\right), \quad (46)$$

$$a'_i = a_i + \delta_i, \quad \delta_i \sim \mathcal{N}\left(0, (\alpha(a_i - \tau))^2\right). \quad (47)$$

Here, the perturbation follows a normal distribution with a mean of 0 and a standard deviation directly proportional to the time difference between the scheduled time and the current time. This specification reflects that later schedule estimates are usually less reliable. α controls the magnitude of perturbation, and three values are tested, denoted by $\alpha_1 = \frac{1}{24}$, $\alpha_2 = \frac{1}{12}$, and $\alpha_3 = \frac{1}{6}$. Given that the farthest flights in the considered horizon are at most 40 min away from the current time τ , the three perturbation settings, according to the 3σ rule, correspond to approximate 99.7% disturbance bounds of ± 5 , ± 10 , and ± 20 min for the farthest flights, respectively. For each value of α , the experiments generate $H = 1000$ sets of independent perturbed schedule times and evaluate how the optimized pushback schedule performs under each perturbed schedule. Here, it is assumed that the flights will try to obey the optimized pushback times as much as possible, unless the updated scheduled times are later than the optimized pushback times. Since an aircraft cannot push back before its ready time, the actual pushback time under this condition is adjusted to the updated scheduled time. Mathematically, the realizable pushback time in perturbation scenario u is defined as

$$x_{i,h}^{*'} = \max\{x_i^*, s'_{i,h}\}. \quad (48)$$

For each perturbed schedule, 1000 independent taxi-time scenarios are sampled, the second-stage problem is solved for each scenario, and the mean queueing time is

calculated. Figure 7 shows the distribution of mean queueing-time values under different perturbation scenarios for each α . The dashed vertical line indicates the unperturbed baseline, whose expected queueing time is 58.39 min.

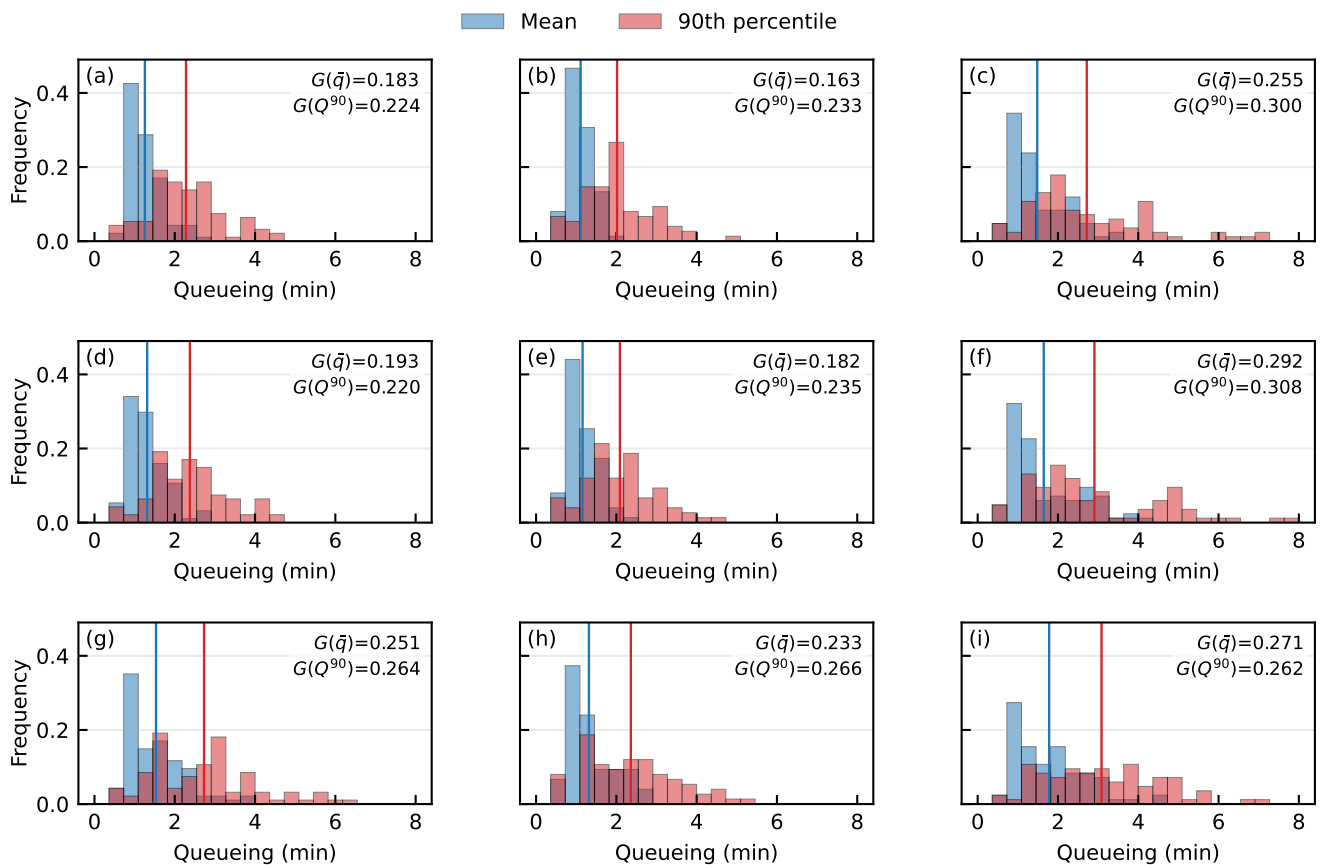


Figure 6. Flight-level distributions of mean and 90th-percentile queueing time. (a) Predicted-distribution SP with sequencing from 10:00–12:00; (b) predicted-distribution SP with sequencing from 12:00–14:00; (c) predicted-distribution SP with sequencing from 16:00–18:00; (d) predicted-distribution SP with FCFS from 10:00–12:00; (e) predicted-distribution SP with FCFS from 12:00–14:00; (f) predicted-distribution SP with FCFS from 16:00–18:00; (g) certainty model with FCFS from 10:00–12:00; (h) certainty model with FCFS from 12:00–14:00; and (i) certainty model with FCFS from 16:00–18:00. The blue and red vertical lines indicate the means of the flight-level mean and 90th-percentile queueing-time distributions, respectively.

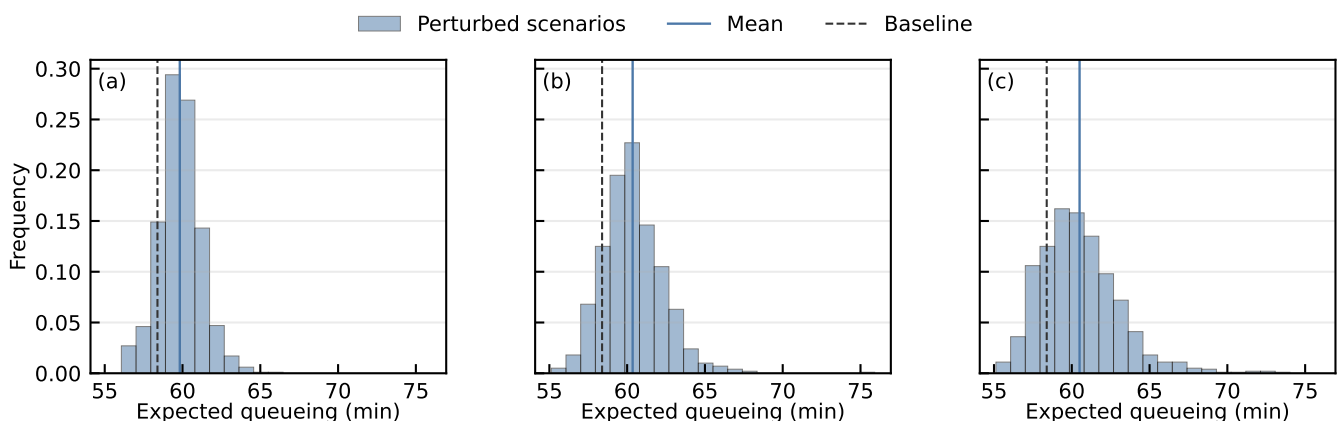


Figure 7. Distribution of expected second-stage FCFS queueing time under scheduled-time perturbations: (a) $\alpha = \alpha_1$; (b) $\alpha = \alpha_2$; and (c) $\alpha = \alpha_3$.

From the figure, it can be seen that after perturbation, the overall queueing level increases. The mean total queueing time increases by about 2 min under each α compared to the baseline. From the perspective of the mean alone, this result is quite optimistic, since the increase is relatively small and not very sensitive to the level of perturbation. However, as α increases, the distribution of expected queueing time becomes more dispersed and the right tail becomes heavier, with the 95th percentile increasing by 3.6–6.4 min (compared to 58.39 min for the baseline).

These results suggest that the local predicted-distribution SP with FCFS schedule remains relatively stable overall, while its tail performance becomes increasingly sensitive to schedule-update perturbations. Most small perturbations are absorbed because the actual pushback rule only delays flights whose updated ready times exceed their optimized pushback times. Under larger values of α , however, more flights become operationally constrained by the updated ready times and the perturbed arrivals more frequently alter runway-availability conflicts. This highlights the value of repeatedly resolving the DM problem in a rolling-horizon implementation: when updated scheduled pushback or arrival times become available, the next sliding-window execution can incorporate them rather than relying only on the earlier fixed plan.

4.7. Practical Implementation Discussion

The proposed framework is designed as an operational decision-support tool for departure metering under the existing airport infrastructure and operational procedures. It could potentially be integrated with Airport Collaborative Decision Making (A-CDM), which enables airport stakeholders to share operational information and coordinate their decisions [30]. By utilizing operational information already exchanged among airport stakeholders, such as estimated off-block times and estimated arrival times, the framework could periodically optimize pushback schedules and provide decision support for departure management. In this study, the optimization is executed every 10 min, and the solution time for each decision horizon is set to 5 min. Such a time scale considers the tradeoff between computational burden and solution quality. Incorporating taxi-time uncertainty requires evaluating the expected cost under various scenarios, inevitably increasing the computational burden compared with the certainty approach. In addition, such a time scale is compatible with practical departure management operations. In operation, departure management decisions should maintain a certain degree of stability, instead of excessively frequent updates [30]. From this perspective, a 10-min optimization cycle combined with a five-minute solution time represents a reasonable balance between solution quality and operational stability. Although some large instances still exhibit nonzero optimality gaps after the solution time, the resulting solutions consistently outperform the certainty approach in terms of queueing reduction, suggesting that meaningful operational benefits can be obtained without requiring the optimization problem to be solved to optimality.

Nevertheless, several implementation challenges remain. The current framework primarily focuses on improving operational efficiency, whereas practical implementation may require additional operational considerations, such as departure punctuality, airline preferences, airport surface vehicle movements, and equity among flights or airlines. These factors also determine whether the proposed DM approach can be applied in real-world operations. Incorporating these factors into the optimization framework while maintaining computational tractability remains an important direction for future research.

5. Conclusions

This study proposed a learning-based two-stage stochastic programming framework for departure metering under taxi-time uncertainty and evaluated its performance using operational data from Tokyo International Airport (RJTT).

The prediction results indicate that static features alone provide sufficient information for practical taxi-time prediction, while incorporating dynamic traffic-state features yields only marginal additional improvement. This finding is particularly important because dynamic traffic-state features depend on pushback decisions, which are themselves optimization variables. By relying only on static features, machine-learning prediction and stochastic optimization can be integrated sequentially without introducing a circular dependency, providing a practical approach for decision-dependent prediction problems. Furthermore, directly predicted flight-level taxi-time distributions provide better probabilistic calibration and coverage than uncertainty distributions constructed from historical point-prediction errors. These results demonstrate that accurately representing flight-specific uncertainty is more informative for stochastic departure metering than relying on a common uncertainty structure estimated statistically from historical prediction errors.

The optimization results demonstrate that explicitly accounting for taxi-time uncertainty consistently improves departure metering performance. Compared with the certainty model, the stochastic models reduce both the typical level and the right-tail risk of runway-area queueing while requiring only limited additional pushback adjustments, demonstrating the operational value of incorporating uncertainty into two-stage decision making. Moreover, the SP model based on directly predicted taxi-time distributions consistently outperforms the error-based SP model, indicating that the quality of uncertainty representation has a direct impact on operational performance. Lower weighted total costs further suggest that queueing reductions are achieved through more efficient operational decisions rather than excessive pushback delays. The equity analysis shows that these improvements are not obtained at the expense of a small subset of flights, while the sensitivity analysis indicates that the proposed framework remains relatively robust under moderate schedule perturbations but benefits from repeated optimization as airport operating conditions evolve.

Overall, these findings suggest that improving uncertainty representation is as important as improving the optimization model itself. More broadly, the results demonstrate that integrating flight-level probabilistic prediction with stochastic optimization provides an effective and practical framework for uncertainty-aware departure metering.

Despite these promising findings, the present study has several limitations that motivate future research. First, this study proposes a general methodological framework for learning-based stochastic departure metering and evaluates its performance using several real-world operating cases at RJTT. Future research should further evaluate the performance of the proposed framework under different airport layouts, operational conditions, and traffic characteristics to provide a more comprehensive assessment of its practical performance. Second, the current framework assumes independent flight-level taxi-time distributions, while correlations among flights caused by shared surface congestion, runway configuration, or weather conditions could be incorporated through richer scenario generation methods. Third, peak-period DM instances can still be difficult to solve to tight optimality gaps; stronger formulations, faster solution strategies, or learning-assisted warm-start strategies may further improve real-time applicability. Fourth, future work may extend the framework to jointly consider arrival management, flight punctuality, surface vehicle movements, and airline equity so that learning-based stochastic departure metering can be evaluated in a broader airport collaborative decision-making environment.

Author Contributions: Conceptualization, C.L. and E.I.; methodology, C.L. and E.I.; software, C.L.; validation, C.L.; formal analysis, C.L. and E.I.; investigation, C.L. and E.I.; resources, E.I.; data curation, C.L.; writing—original draft preparation, C.L.; writing—review and editing, C.L. and E.I.; visualization, C.L.; supervision, E.I.; project administration, E.I.; funding acquisition, E.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by JSPS KAKENHI Grant Numbers 24KK0088 and 25K07952.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available because the source data are commercially available by a data vendor.

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

Abbreviations

The following abbreviations are used in this manuscript:

DM	Departure metering
FCFS	First-come, first-served
ML	Machine learning
SAA	Sample average approximation
SG	Scenario generation
SP	Stochastic programming

Appendix A. Scenario Generation Model

This appendix provides the integer programming formulation used to generate the representative taxi-time scenarios in Section 3.3.2. The model follows the p-median scenario-reduction logic used in stochastic air-traffic scheduling, where a large empirical sample is compressed into a smaller set of representative scenarios and each original sample is assigned to one selected representative scenario [31].

Let $\mathcal{N} = \{1, \dots, N\}$ denote the initial SAA sample set. Each sample $n \in \mathcal{N}$ corresponds to a taxi-time vector $\tilde{\mathbf{T}}^n = (\tilde{T}_i^n)_{i \in \mathcal{F}^{\text{dep}}}$ and has initial probability $P_0^n = 1/N$ for each scenario. The goal is to select L representative scenarios from $\mathcal{N}$, where $L \ll N$. For each pair of samples (n, m) , the dissimilarity between their taxi-time vectors is measured by the weighted L_1 distance

$$d_{n,m} = P_0^n \sum_{i \in \mathcal{F}^{\text{dep}}} \omega_i |\tilde{T}_i^n - \tilde{T}_i^m|, \quad \forall n, m \in \mathcal{N}. \quad (\text{A1})$$

The decision variables are defined as follows:

μ_m : binary variable equal to 1 if sample m is selected as a representative scenario, and 0 otherwise, $\forall m \in \mathcal{N}$.

$v_{n,m}$: binary variable equal to 1 if original sample n is assigned to representative scenario m , and 0 otherwise, $\forall n, m \in \mathcal{N}$.

The scenario generation model is formulated as

$$\text{SG: } \min_{\mu, v} \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{N}} d_{n,m} v_{n,m}, \quad (\text{A2})$$

$$\text{s.t. } \sum_{m \in \mathcal{N}} \mu_m = L, \quad (\text{A3})$$

$$v_{n,m} \leq \mu_m, \quad \forall n, m \in \mathcal{N}, \quad (\text{A4})$$

$$\sum_{m \in \mathcal{N}} v_{n,m} = 1, \quad \forall n \in \mathcal{N}, \quad (\text{A5})$$

$$\mu_m \in \{0, 1\}, \quad \forall m \in \mathcal{N}, \quad (\text{A6})$$

$$v_{n,m} \in \{0, 1\}, \quad \forall n, m \in \mathcal{N}. \quad (\text{A7})$$

The objective function (A2) minimizes the total probability-weighted distance from the original SAA sample set to the selected representative scenarios. Constraint (A3) requires exactly L representative scenarios. Constraint (A4) ensures that an original sample can only be assigned to a selected representative scenario. Constraint (A5) assigns each original sample to exactly one representative scenario. Constraints (A6) and (A7) define the binary domains.

After solving SG, the selected samples are relabeled as representative scenarios $l = 1, \dots, L$. The reduced scenario set is

$$\{\tilde{T}^1, \dots, \tilde{T}^L\} = \{\tilde{T}^m : \mu_m^* = 1, m \in \mathcal{N}\}. \quad (\text{A8})$$

Let $m(l)$ denote the original sample index corresponding to representative scenario l . The probability of each selected scenario is obtained by aggregating the probabilities of the original samples assigned to it:

$$P^{(l)} = \sum_{n \in \mathcal{N}} P_0^n v_{n,m(l)}^*, \quad \forall l = 1, \dots, L. \quad (\text{A9})$$

The resulting scenario-probability pairs $\{(\tilde{T}^l, P^{(l)})\}_{l=1}^L$ are then used in SP-SG (31). In the DM implementation, the same SG formulation is solved for each DM execution using only the flights included in $\mathcal{W}_\tau$, producing $\{(\tilde{T}_\tau^l, P_\tau^{(l)})\}_{l=1}^L$ with the same number of representative scenarios in each instance.

Appendix B. Distribution Prediction Metrics

This appendix defines the metrics used to evaluate taxi-time distribution predictions in Section 3.1. Let N denote the number of test flights and K denote the number of taxi-time bins. For flight i , the realized taxi time is t_i , and the predicted probability of bin k is $p_{i,k}$, where $\sum_{k=1}^K p_{i,k} = 1$. Let c_k , l_k , and u_k denote the representative time, lower bound, and upper bound of bin k , respectively. In the implementation, 10-s bins are used and c_k is the midpoint of bin k . Let k_i be the bin index containing t_i .

The LogLoss is calculated as

$$\text{LogLoss} = -\frac{1}{N} \sum_{i=1}^N \log(p_{i,k_i}). \quad (\text{A10})$$

For CRPS, define the predicted cumulative distribution and realized step function as

$$F_{i,k} = \sum_{m=1}^k p_{i,m}, \quad H_{i,k} = \mathbb{I}\{k \geq k_i\}, \quad (\text{A11})$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. With bin width $\Delta = 10$ s, the discrete CRPS is computed as

$$\text{CRPS} = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K (F_{i,k} - H_{i,k})^2 \Delta. \quad (\text{A12})$$

The 90% central prediction interval is obtained from the 5% and 95% predicted quantiles. Let $L_i^{0.90}$ and $U_i^{0.90}$ denote its lower and upper bounds. Then PICP and MPIW are calculated as

$$\text{PICP} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}\{L_i^{0.90} \leq t_i \leq U_i^{0.90}\}, \quad (\text{A13})$$

$$\text{MPIW} = \frac{1}{N} \sum_{i=1}^N (U_i^{0.90} - L_i^{0.90}). \quad (\text{A14})$$

Appendix C. Optimization Model Parameter Settings

Table A1 summarizes the numerical parameter values used in the SP-SG optimization model. The predicted distribution and error-based distribution use the same optimization model parameters; they differ only in how the taxi-time scenarios are generated before applying SG.

Table A1. Parameter settings used in the SP-SG implementation.

Parameter	Value	Description
Maximum pushback delay d	5 min	Upper bound on the first-stage delay variable for each departure flight.
Minimum runway-area queueing time $\underline{q}$	40 s	Lower bound on q_i for every departure flight and scenario.
Maximum runway-area queueing time $\bar{q}$	1200 s/900 s	Upper bound on q_i in the FCFS and sequencing experiments, respectively.
Pushback delay penalty λ	0.2	Objective coefficient for total pushback delay in seconds.
Arrival runway occupation time u_k	120 s	Time interval before the scheduled arrival runway-exit time blocked for departure conflicts.
Post-arrival buffer time v_k	30 s	Time interval after the scheduled arrival runway-exit time blocked for departure conflicts.
Minimum departure separation $\Delta_{i,j}$	Table A2	Wake-category-dependent separation time for two departure flights using the same runway, where i leads j .

Table A2. Wake-separation settings for departure–departure pairs.

Leading Flight	Trailing Flight		
	H	M	L
H	60 s	120 s	180 s
M	60 s	60 s	120 s
L	60 s	60 s	60 s

For any departure–departure pair $(i, j) \in \mathcal{I}^{\text{DD}}$, the separation value $\Delta_{i,j}$ is selected from Table A2 according to the wake categories of the leading flight i and the trailing flight j . If a wake category is missing or not one of H, M, or L in the input data, it is treated as L in the implementation. The prepared data first store the separation as a number of 10-s slices using a ceiling operation, and the SP-SG scripts convert it back to seconds before solving the optimization model.

Appendix D. Supplementary Results for Additional Operating Days

To examine the results on additional operating days, the rolling DM experiment was also conducted for 9 May 2023 and 8 July 2023. 9 May 2023 is a typical north-wind-operation day and 8 July 2023 is a typical south-wind-operation day defined in [1]. The same predicted taxi-time distribution, FCFS second-stage queueing rule, rolling-window configuration, and SP-SG settings ($N = 1000$ and $L = 100$) as in the main experiment were used. Figure A1

summarizes the results. Each row corresponds to one operating day. The first column reports the Gurobi gap of each solved rolling-window instance, with marker color indicating the number of departures included in that instance; consistent with Figure 3. The second column reports the hourly total pushback displacement. The third column compares the distributions of total second-stage queueing time produced by the predicted-distribution SP and certainty models for the 10:00–12:00 test period under 1000 independently taxi-time scenarios sampled from the predicted distribution. The horizontal dotted lines in the first column denotes the 95th-percentile gaps, and the vertical lines in the third column denote the mean queueing times.

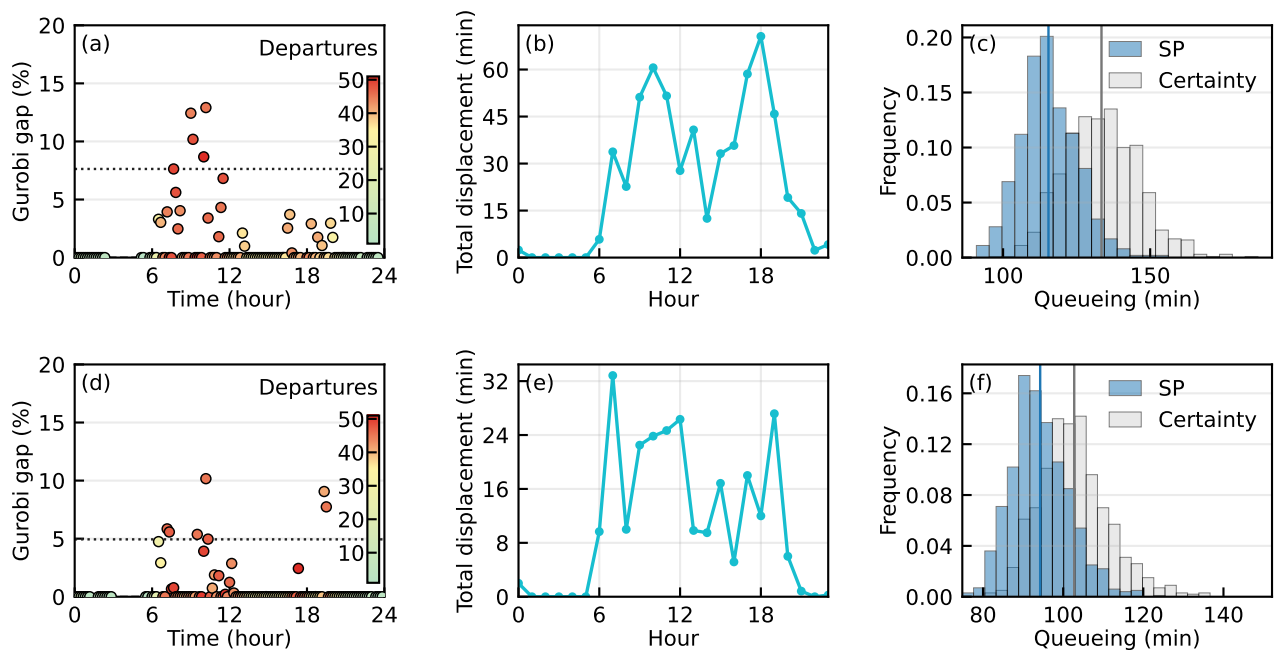


Figure A1. Supplementary results for two additional operating days using the predicted distribution and FCFS second-stage queueing: (a) rolling-window solution gaps and problem sizes on 9 May 2023; (b) hourly total pushback displacement on 9 May 2023; (c) 10:00–12:00 test-set queueing-time distributions on 9 May 2023; (d) rolling-window solution gaps and problem sizes on =8 July 2023; (e) hourly total pushback displacement on =8 July 2023; and (f) 10:00–12:00 test-set queueing-time distributions on =8 July 2023. In (a) and (d), marker color indicates the number of departures, while dashed and dotted horizontal lines indicate the median and 95th-percentile gaps, respectively. In (b) and (e), the cyan lines indicate hourly total pushback displacement. In (c) and (f), the blue and gray vertical lines indicate the means of the predicted-distribution SP and certainty-model distributions, respectively.

The results show that the proposed framework exhibits similar overall performance on both additional operating days, although the magnitude of the effects differs between them. For the rolling optimization, the median Gurobi gap is 0 on both days, and 92.6% of the 9 May instances and 95.1% of the 8 July instances finish within a 5% gap. The 95th-percentile gaps are 7.6% and 4.9%, respectively. The relatively few high-gap instances generally coincide with larger rolling-window problems, consistent with the relationship between traffic density and convergence difficulty observed on the main test day.

The pushback displacement are more concentrated in the late-morning and evening traffic peaks and reaching approximately 61 min at 10:00 and 71 min at 18:00. By comparison, the 8 July displacement profile is lower and more dispersed across the day, with its largest hourly value of approximately 33 min occurring at 7:00.

In the independent 10:00–12:00 evaluation, the predicted-distribution SP shifts the total queueing-time distribution to the left on both days. Relative to the certainty model, it reduces the mean queueing time from 133.4 to 115.4 min on 9 May and from 102.8 to 94.3 min on 8 July, corresponding to reductions of 13.5% and 8.3%, respectively. The 95th-percentile queueing time is also reduced from 154.3 to 131.3 min and from 118.5 to 107.0 min, showing that the benefit is present not only in the average outcome but also in the right tail. These results demonstrate that the proposed framework remains computationally applicable under both north- and south-wind operations and provides queueing reductions beyond the main test day.

References

1. Kato, F.; Itoh, E. Aircraft Taxi Time Prediction Using Machine Learning and its Application for Departure Metering. In *Proceedings of the 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*; IEEE: New York, NY, USA, 2023; pp. 1–9.
2. Wang, X.; Brownlee, A.E.I.; Woodward, J.R.; Weiszer, M.; Mahfouf, M.; Chen, J. Aircraft Taxi Time Prediction: Feature Importance and Their Implications. *Transp. Res. Part C Emerg. Technol.* **2021**, *124*, 102892. <https://doi.org/10.1016/j.trc.2020.102892>.
3. Vakaet, C. Taxi Time Prediction with Classical and Auto Machine Learning at Schiphol Airport. Master's Thesis, Delft University of Technology, Delft, The Netherlands, 2021.
4. Herrema, F.; Curran, R.; Visser, H.; Huet, D.; Lacote, R. Taxi-Out Time Prediction Model at Charles de Gaulle Airport. *J. Aerosp. Inf. Syst.* **2018**, *15*, 120–130. <https://doi.org/10.2514/1.1010502>.
5. Lian, G.; Zhang, Y.; Desai, J.; Xing, Z.; Luo, X. Predicting Taxi-Out Time at Congested Airports with Optimization-Based Support Vector Regression Methods. *Math. Probl. Eng.* **2018**, *2018*, 7509508. <https://doi.org/10.1155/2018/7509508>.
6. Yin, J.; Hu, Y.; Ma, Y.; Xu, Y.; Han, K.; Chen, D. Machine Learning Techniques for Taxi-Out Time Prediction with a Macroscopic Network Topology. In *Proceedings of the 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, London, UK, 23–27 September 2018; pp. 1–8. <https://doi.org/10.1109/DASC.2018.8569664>.
7. Murça, M.C.R. A Robust Optimization Approach for Airport Departure Metering under Uncertain Taxi-Out Time Predictions. *Aerosp. Sci. Technol.* **2017**, *68*, 269–277. <https://doi.org/10.1016/j.ast.2017.05.020>.
8. Badrinath, S.; Balakrishnan, H.; Joback, E.; Reynolds, T.G. Impact of Off-Block Time Uncertainty on the Control of Airport Surface Operations. *Transp. Sci.* **2020**, *54*, 920–943. <https://doi.org/10.1287/trsc.2019.0957>.
9. Chen, H.; Solak, S. Lower Cost Departures for Airlines: Optimal Policies under Departure Metering. *Transp. Res. Part C Emerg. Technol.* **2020**, *111*, 531–546. <https://doi.org/10.1016/j.trc.2019.12.023>.
10. Chandra, A.; Choubey, N.; Verma, A.; Sooraj, K.P. Quasi-Stochastic Departure Metering Model Considering Non-Deterministic Taxi Time, Standard Instrument Departure Track Time, and a New Delay–Conflict Relationship. *Transp. Res. Rec.* **2024**, *2678*, 341–358. <https://doi.org/10.1177/03611981241240761>.
11. Kang, J.; Bao, J.; Zhang, J.; Tang, X.; Han, J.; He, J. A Two-Stage Stochastic Optimization Approach for Mega-Airport Departure Metering under Data-Driven Taxi-Time Uncertainty Predictions. *J. Air Transp. Manag.* **2026**, *133*, 102973. <https://doi.org/10.1016/j.jairtraman.2026.102973>.
12. Simaiakis, I.; Sandberg, M.; Balakrishnan, H. Dynamic Control of Airport Departures: Algorithm Development and Field Evaluation. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 285–295. <https://doi.org/10.1109/TITS.2013.2278484>.
13. Sandberg, M.; Simaiakis, I.; Balakrishnan, H.; Reynolds, T.G.; Hansman, R.J. A Decision Support Tool for the Pushback Rate Control of Airport Departures. *IEEE Trans. Hum.-Mach. Syst.* **2014**, *44*, 416–421. <https://doi.org/10.1109/THMS.2014.2305906>.
14. Simaiakis, I.; Khadilkar, H.; Balakrishnan, H.; Reynolds, T.G.; Hansman, R.J. Demonstration of Reduced Airport Congestion through Pushback Rate Control. *Transp. Res. Part A Policy Pract.* **2014**, *66*, 251–267. <https://doi.org/10.1016/j.tra.2014.05.014>.
15. Malik, W.; Gupta, G.; Jung, Y.C. Managing Departure Aircraft Release for Efficient Airport Surface Operations. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, ON, Canada, 2–5 August 2010. <https://doi.org/10.2514/6.2010-7696>.
16. Malik, W.; Gupta, G.; Jung, Y.C. Spot Release Planner: Efficient Solution for Detailed Airport Surface Traffic Optimization. In *Proceedings of the 12th AIAA Aviation Technology, Integration, and Operations Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, IN, USA, 17–19 September 2012.
17. Bertsimas, D.; Frankovich, M. Unified Optimization of Traffic Flows Through Airports. *Transp. Sci.* **2016**, *50*, 77–93. <https://doi.org/10.1287/trsc.2015.0590>.
18. Badrinath, S.; Balakrishnan, H.; Ma, J.; Delahaye, D. Comparative Analysis of Departure Metering at United States and European Airports. *J. Air Transp.* **2020**, *28*, 93–104. <https://doi.org/10.2514/1.D0179>.

19. Ali, H.; Pham, D.T.; Alam, S.; Schultz, M. A Deep Reinforcement Learning Approach for Airport Departure Metering Under Spatial–Temporal Airside Interactions. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 23933–23950. <https://doi.org/10.1109/TITS.2022.3209397>.
20. Ali, H.; Pham, D.T.; Alam, S. Toward Greener and Sustainable Airside Operations: A Deep Reinforcement Learning Approach to Pushback Rate Control for Mixed-Mode Runways. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 18354–18367. <https://doi.org/10.1109/TITS.2024.3433428>.
21. Pujet, N.; Delcaire, B.; Feron, E. Input–Output Modeling and Control of the Departure Process of Congested Airports. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Portland, OR, USA, 9–11 August 1999; pp. 1835–1852. <https://doi.org/10.2514/6.1999-4299>.
22. Idris, H.R.; Delcaire, B.; Anagnostakis, I.; Hall, W.D.; Pujet, N.; Feron, E.; Hansman, R.J.; Clarke, J.P. Observations of Departure Processes at Logan Airport to Support the Development of Departure Planning Tools. *Air Traffic Control Q.* **1999**, *7*, 229–257. <https://doi.org/10.2514/atcq.7.4.229>.
23. Simaiakis, I.; Balakrishnan, H. A Queuing Model of the Airport Departure Process. *Transp. Sci.* **2016**, *50*, 94–109. <https://doi.org/10.1287/trsc.2015.0603>.
24. Itoh, E.; Mitici, M.; Schultz, M. Modeling Aircraft Departure at a Runway Using a Time-Varying Fluid Queue. *Aerospace* **2022**, *9*, 119. <https://doi.org/10.3390/aerospace9030119>.
25. Lee, H.; Malik, W.; Zhang, B.; Nagarajan, B.; Jung, Y.C. Taxi Time Prediction at Charlotte Airport Using Fast-Time Simulation and Machine Learning Techniques. In Proceedings of the 15th AIAA Aviation Technology, Integration, and Operations Conference, Dallas, TX, USA, 22–26 June 2015. <https://doi.org/10.2514/6.2015-2272>.
26. Lim, Y.; Alam, S.; Lilith, N.; Tan, F.; Toon, P.L. Graph Neural Networks with Spatiotemporal Flow Features for Aircraft Taxi-Out-Time Prediction. *J. Air Transp.* **2025**, *33*, 297–315. <https://doi.org/10.2514/1.D0369>.
27. Kleywegt, A.J.; Shapiro, A.; Homem-de Mello, T. The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM J. Optim.* **2002**, *12*, 479–502. <https://doi.org/10.1137/S1052623499363220>.
28. Heitsch, H.; Romisch, W. Scenario Reduction Algorithms in Stochastic Programming. *Comput. Optim. Appl.* **2003**, *24*, 187–206. <https://doi.org/10.1023/A:1021805924152>.
29. Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*; Gurobi Optimization, LLC: Beaverton, OR, USA, 2026.
30. International Air Transport Association. *Airport Collaborative Decision Making (A-CDM) Toolkit*; International Air Transport Association: Montreal, QC, Canada, 2025.
31. Wang, K.; Jacquillat, A. A Stochastic Integer Programming Approach to Air Traffic Scheduling and Operations. *Oper. Res.* **2020**, *68*, 1375–1402. <https://doi.org/10.1287/opre.2020.1985>.

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