# A Stochastic Integer Programming Approach to Air Traffic Scheduling and Operations

Kai Wang
Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, chwangkai@gmail.com

Alexandre Jacquillat
Heinz College of Information Systems and Public Policy, Carnegie Mellon University, ajacquil@andrew.cmu.edu

Air traffic management measures comprise tactical operating procedures to minimize delay costs, and strategic scheduling interventions to control over-capacity scheduling. Although interdependent, these problems have been treated in isolation. This paper proposes an *Integrated Model of Scheduling and Operations in Airport Networks* that jointly optimizes scheduling interventions and ground-holding operations across airports networks, under operating uncertainty. It is formulated as a two-stage stochastic program with integer recourse. To solve it, we develop an original decomposition algorithm with provable quality guarantees. The algorithm relies on new optimality cuts—*dual integer cuts*—which leverage the reduced costs of the dual linear programming relaxation of the second-stage problem. The algorithm also incorporates *neighborhood constraints*, which shift from exploration to exploitation at later stages. Moreover, we propose a data-driven scenario generation procedure that constructs representative scenarios for stochastic programming from historical records of operations. Computational experiments show that our algorithm yields near-optimal solutions for networks of the size of the US National Airspace System. Ultimately, the proposed approach enhances airport demand management models through scale integration (by capturing network-wide interdependencies) and scope integration (by capturing interdependencies between scheduling and operations).

*Key words*: Airport demand management; Stochastic integer programming; Benders decomposition; Benefits of integration.

*History*: Submitted November 2018; revision received June 2019

## 1. Introduction

Air traffic management systems have been facing strong demand growth over decades, while available infrastructure has remained limited. Resulting imbalances between demand and capacity can lead to severe congestion and significant costs. For instance, the nation-wide impact of flight delays in the United States in 2007 was estimated at over $30 billion (Ball et al. 2010). Absent capacity expansion opportunities, congestion mitigation levers fall into two broad categories: (i) tactical interventions aimed to enhance operations in air traffic networks, and (ii) strategic demand management interventions aimed to align flight schedules with airports' operating capacities.

At the tactical level, air traffic control and air traffic flow management interventions aim to optimize aircraft operations and minimize delay costs—these are implemented during each day of operations. One of the most critical interventions is known as ground holding, which involves

controlling flights' departures to absorb delays at departure airports rather than in the air and mitigate their economic, safety and environmental costs. These initiatives have been successfully implemented in the US and European airspaces. Nonetheless, they cannot prevent high delays when the number of flights scheduled at busy airports exceeds available capacity by a wide margin.

At the strategic level, scheduling interventions refer to the demand management rules that govern airline scheduling—these are implemented months in advance of operations. Outside the United States, busy airports are subject to "schedule coordination": they declare a value of capacity (usually set conservatively close to poor-weather capacity), and allocate a corresponding number of slots to the airlines through an administrative procedure. The use of demand management is much more limited at US airports. Only a few of the busiest airports are subject to "flight caps", which are less stringent than declared capacities set at comparable schedule-coordinated airports; flight schedules at other airports are subject to no restrictions. These variations suggest a trade-off between schedule coordination, which places a premium on congestion mitigation, and *laissez-faire*, which places a premium on high scheduling levels (de Neufville and Odoni 2013).

Existing approaches to optimize scheduling interventions and air traffic operations have two main shortcomings. First, most models of scheduling interventions focus on a single airport; yet, flight schedules exhibit interdependencies across airport networks. Second, the strategic scheduling problem and the tactical operating problem have been mainly treated in isolation; yet, they are interdependent. Air traffic operations obviously depend on flight schedules; vice versa, the more effective air traffic operations, the less aggressive scheduling interventions need to be to guarantee any level of service. These interdependencies are complicated by uncertainties at the time of scheduling regarding the weather and other conditions that will prevail at the time of operations. Thus, the schedule cannot simply match operating capacity with certainty across spatial-temporal networks. Ultimately, this creates a trade-off between absorbing demand-capacity imbalances at the strategic level through scheduling interventions vs. at the tactical level through flight delays.

This paper proposes an original *integrated* approach to air traffic congestion mitigation that jointly optimizes strategic scheduling interventions and tactical ground-holding operations in *networks* of capacity-constrained airports, under operating *uncertainty*. This approach augments prior models of scheduling interventions through scale integration (i.e., by capturing network-wide interdependencies) and scope integration (i.e., by capturing interdependencies between strategic scheduling and tactical operations). Note, importantly, that this paper does *not* aim to replace all models of scheduling interventions and air traffic operations with a single unified model. Instead, the proposed integrated model aims to capture the stochasticity and the endogeneity of ground-holding operations when optimizing scheduling interventions—that is, how airport operations will respond to any schedule of flights in various operating scenarios. For any resulting schedule of

flights, flight operations can then be optimized using state-of-the-art models of air traffic control and flow management (see Ball et al. 2007, Vossen et al. 2012, for excellent reviews).

Specifically, this paper makes the following five contributions.

From a modeling standpoint (Section 3), we formulate an *Integrated Model of Scheduling and Operations in Airport Networks* (IMSOAN) as a bi-objective two-stage stochastic integer program. The first stage optimizes scheduling interventions to minimize deviations from airlines' preferences, subject to scheduling and network connectivity constraints. These decisions are made before operating uncertainty is resolved, thus relying on probabilistic characterizations of air traffic operations. The second stage considers the multi-airport ground-holding problem, which optimizes operations in airport networks to minimize delay costs, subject to operating, network connectivity, and airport capacity constraints. These decisions are made as information on operating conditions becomes available. The (IMSOAN) results in a very large-scale two-stage stochastic integer program.

From a methodological standpoint (Section 4), this paper develops a new decomposition algorithm with provable solution quality guarantees for two-stage stochastic programs with integer recourse—a class of problems that is notoriously challenging to solve. The proposed algorithm relies on new optimality cuts, named *dual integer cuts*, that leverage the reduced costs of the dual linear programming relaxation of the second-stage problem. These cuts tighten the traditional Benders cuts in two-stage stochastic programs (with continuous or integer recourse). At each iteration, the algorithm solves the linear programming relaxation of the second-stage problem, which enable the implementation of acceleration techniques from stochastic continuous programming based on local branching and Pareto-optimality cuts. We also propose a novel acceleration approach using *neighborhood constraints*, which restrict the search in later iterations to the neighborhood of the solutions visited in initial iterations—thus shifting from exploration to exploitation. Ultimately, our dual integer cuts and neighborhood constraints improve the convergence of decomposition algorithms—especially when the first-stage problem admits a large solution space.

From an empirical standpoint (Section 5), we propose a data-driven scenario generation procedure that constructs representative scenarios and their probabilities—a common challenge in stochastic programming. Instead of making distributional assumption on uncertain parameters (weather conditions, in our case), this paper generates scenarios from historical data. Specifically, the proposed procedure starts from a large sample of scenarios observed empirically, selects a subset of them, and maps each original scenario to one of the selected scenarios. This problem is cast as a $p$-median problem (Reese 2006) by minimizing the distance from the original scenario set to the set of selected scenarios, subject to cover and budget constraints.

From a computational standpoint (Section 6), we obtain high-quality solutions for networks of the size of the US National Airspace System. In our largest test instances—involving 8.2 million

integer variables and 20.8 million constraints—direct CPLEX implementation does not even yield a feasible solution, whereas our algorithm yields near-optimal solutions in reasonable computational times. Results also show the benefits of our stochastic integer programming algorithm. First, the dual integer cuts significantly tighten traditional Benders cuts—by reducing the convergence gap from 20% to 1%. Second, the neighborhood constraints improve the algorithm's convergence, as compared to implementations based only on existing acceleration techniques (i.e., local branching and Pareto-optimality cuts). Third, the value of the stochastic solution is very large, thus underscoring the benefits of our stochastic optimization approach. Fourth, our scenario generation approach significantly reduces expected costs, as compared to heuristic procedures.

From a managerial standpoint (Section 7), results demonstrate the benefits of scale and scope integration, i.e., the benefits of capturing scheduling interdependencies across full-scale airport networks (as compared to partial approaches) and the benefits of capturing interdependencies between scheduling and operations (as compared to sequential approaches). Moreover, (IMSOAN) provides transparent decision-making support to inform to which extent, where (i.e., at which airports) and when (i.e., at which times) to consider congestion-mitigating scheduling adjustments. First, rescheduling even 1% of flights by 15 minutes each can reduce expected network-wide delays by 20%–30%. Second, optimal scheduling interventions adjust flight schedules throughout the network but are concentrated at busier airports to minimize system-wide delays. Third, optimal scheduling interventions are concentrated at earlier times of the day to avoid the formation and propagation of delays, while most demand-capacity imbalances are absorbed through flight delays at later times.

Ultimately, the proposed modeling and computational framework enhances existing demand management approaches by balancing the strategic costs of scheduling interventions and the tactical costs of delays across air traffic networks. At schedule-coordinated airports, it can support the setting of declared capacities and network-wide slot allocation. This can augment existing decentralized practices where (i) capacity declaration is left up to each airport, leading to a lack of standardization, and (ii) slot allocation is performed at each airport separately and conflicts are resolved in an *ad hoc* manner at bi-annual slot conferences. At US airports, this paper shows that limited and targeted scheduling interventions can lead to significant network-wide delay reductions.

## 2. Literature Review

This paper contributes to the literatures on air traffic management, stochastic integer programming, and scenario reduction. We briefly review them in this section.

### 2.1. Air Traffic Management

The tactical interventions aimed to optimize aircraft flows across capacity-constrained networks of operations are referred to as air traffic flow management (ATFM). One of the major ATFM levers

involves the ground-holding flights at the departure airport—where delay costs are lower than in the terminal airspace at the arrival airport. Odoni (1987) first formalized this problem. Richetta and Odoni (1993) and Terrab and Odoni (1993) studied the single-airport ground-holding problem, i.e., the optimization of the departure times of all flights scheduled to arrive at one airport. Vranas et al. (1994) considered the multi-airport ground-holding problem, i.e., the optimization of flights' departure and arrival times across multiple airports. ATFM solutions were further enhanced by capturing capacity-constrained en-route operations (Bertsimas and Stock Patterson 1998), aircraft routing (Bertsimas et al. 2011), and operating uncertainty (Ball et al. 2003, Mukherjee and Hansen 2007, Jones et al. 2018). In this paper, the second-stage model of air traffic operations reflects the multi-airport ground-holding problem, thus assuming uncapacitated en-route operations—motivated by the fact that busy airports operate as the main bottlenecks of air traffic operations.

Strategic airport demand management has also attracted significant attention in economics and operations research (see Czerny et al. 2008, Gillen et al. 2016, for reviews). Market-based mechanisms based on congestion pricing (Carlin and Park 1970) and slot auctions (Rassenti et al. 1982) have been proposed. In practice, however, solutions involving monetary transfers have not been widely implemented. This paper focuses on existing non-monetary scheduling mechanisms.

Recent models and algorithms have optimized scheduling interventions at a single airport. At schedule-coordinated airports, (Zografos et al. 2012, Ribeiro et al. 2017, 2019) developed integer programs to support slot allocation by minimizing deviations from airlines' requests, given the administrative rules in place. Pyrgiotis and Odoni (2016) replicate the effects of scheduling limits at busy US airports, showing that limited scheduling interventions could lead to large delay reductions. Other objectives have also been considered, such as acceptability by the airlines (Zografos et al. 2017) and equity (Jacquillat and Vaze 2018, Jiang and Zografos 2018).

Closely related to our paper, Jacquillat and Odoni (2015) jointly optimize airport scheduling interventions and operating procedures (i.e., the selection of runway configurations and the balancing of arrivals and departures). This approach also balances the strategic costs of scheduling interventions and the tactical costs of delays. However, it focuses on a single airport, thus failing to capture interdependencies across multiple airports. Our paper addresses this shortcoming by explicitly considering network-wide effects in scheduling interventions and resulting air traffic operations. It thus integrates network-wide operating dynamics—captured through the multi-airport ground-holding problem—rather than replicating fine-grained procedures at any single airport into the optimization of scheduling interventions. From a technical standpoint, this results in a different modeling and algorithmic framework based on stochastic integer programming.

Finally, recent models aim to optimize slot allocation in a network of airports. An initial solution was provided by Pellegrini et al. (2012) using local search heuristics. Pellegrini et al. (2017) improved it using exact methods. Following schedule coordination practices, they considered declared

capacities but did not account for the interdependencies between scheduling and operations. Coroni et al. (2014) made the only attempt to trade off network-wide costs of scheduling interventions and flight delays. However, they rely on an aggregate model of air traffic operations that does not capture flight-level operating dynamics. Moreover, they apply the model to a subset of the European network, which is significantly smaller than the full US network considered in our paper.

### 2.2. Stochastic Integer Programming

Benders decomposition has enabled considerable progress in stochastic programming with continuous recourse (Benders 1962) by iterating between a "master problem" and continuous "subproblems". However, this approach is not designed to accommodate stochastic integer programs, which generally have non-convex and non-continuous recourse functions (Schultz 1993).

Louveaux and Schultz (2003) and Sen (2005) provide comprehensive surveys on stochastic integer programming. Several decomposition approaches have been proposed, based on enumeration (Schultz et al. 1998), branch and bound (Ahmed et al. 2004), branch and cut (Sen and Sherali 2006), Gomory cuts (Zhang and Kucukyavuz 2014), split cuts (Sen and Higle 2005), sample average approximation (Kleywegt et al. 2001), disjunctive programming (Ntaimo 2010) and dynamic programming (Zou et al. 2017). The ideas of Benders decomposition have been extended to integer recourse functions using Lagrangian relaxation and branch-and-bound algorithms, but the resulting master problem is non-convex (Caroe and Schultz 1997). One of the most widely applied solution approaches is the integer L-shaped method from Laporte and Louveaux (1993). This method generates optimality and feasibility cuts iteratively to the master problem by leveraging the optimal value of the second-stage objective function for any first-stage integer solution. This method was extended with non-linear and Gomory cuts (Caroe and Schultz 1999), with specialized branching (Ahmed et al. 2004), with thin-direction branching and multicut heuristics (Kim and Mehrotra 2015), and with alternating cuts and a cut-generating optimization model (Angulo et al. 2016).

These developments have enabled applications of stochastic integer programming to vehicle routing (Laporte et al. 2002), capacity expansion (Ahmed and Sahinidis 2003), inventory management (Gunpinar and Centeno 2015), nurse staffing and scheduling (Kim and Mehrotra 2015), etc. At the same time, the integer L-shaped method is most effective when the first-stage problem has a limited solution space. Our problem in this paper, however, admits an exponentially large first-stage solution space—so that the integer L-shaped algorithm did not converge even in small instances.

These challenges motivate new solution algorithms for stochastic integer programming. We propose a novel decomposition approach that leverages the dual linear programming relaxation of the second-stage integer problem and its reduced costs. As such, our approach shares similarities with the additive bounding procedure from Fischetti and Toth (1989, 1993), which uses dual information from a model relaxation to derive a lower bound for NP-hard combinatorial optimization

problems—with applications to the traveling salesman and vehicle routing problems (Fischetti and Toth 1988, 1992, Baldacci et al. 2008, Baldacci and Mingozzi 2009). In contrast, our paper leverages such information to derive new optimality cuts, which are valid for a broad class of two-stage stochastic (continuous or integer) programs in which the second-stage variables can be bounded below by linear (or convex) terms involving first-stage variables.

### 2.3. Scenario Reduction

A challenge in stochastic programming involves defining uncertainty scenarios and their probabilities. Most applications rely on distributional assumptions regarding the uncertain parameters to sample scenarios. Scenario reduction methods have been proposed when the number of scenarios is too large to ensure computational tractability (Dupačová et al. 2003, Römisch 2009). These methods start with a "true" probability distribution, and generate another probability distribution of prescribed cardinality that approximates it best. The scenario reduction problem is typically solved using heuristic algorithms. It has been applied, for instance, to the unit commitment and capacity expansion problems in energy planning (Carrión et al. 2007, Morales et al. 2009). In contrast, the scenario generation approach developed in this paper makes no distributional assumption but, instead, constructs representative scenarios and their probabilities from historical data.

## 3. Model of Scheduling and Operations in Airport Networks

We describe the Integrated Model of Scheduling and Operations in Airport Networks (IMSOAN) in Section 3.1 and formulate it mathematically in Section 3.2. Section 3.3 discusses its complexity.

### 3.1. Model Description

(IMSOAN) takes as inputs: (i) airlines' preferred schedules of flights and planned aircraft and passenger connections, (ii) capacity estimates at each airport, and (iii) operating scenarios. Scheduling inputs can be obtained from airlines' scheduling requests.[1] Airport capacity is represented by means of a convex piecewise-linear *capacity envelope*, which specifies the non-increasing relationship between the number of arrivals and departures that can be operated per period of time (Gilbo 1993), and is constructed from historical data. At each airport, we define one envelope per time period, based on the runway configuration (i.e., the set of active runways to operate arrivals and departures) and weather conditions. Scenarios characterize the operating conditions (e.g., weather) that will prevail, and their respective probabilities (see our scenario generation procedure in Section 5). Scenario realizations determine the capacity of each airport at each time of day.

[1] The mechanism through which airlines submit scheduling requests lies beyond the scope of this paper. In particular, passenger connections data may be unavailable to schedule coordination entities. Still, these can be reconstructed using data analytics (see, e.g., Barnhart et al. 2014) or provided by the airlines in a collaborative environment.

(IMSOAN) optimizes strategic scheduling interventions (SI) in the first stage and tactical multi-airport ground-holding operations (MAGHO) in the second stage, across the network of capacitated airports. The first stage determines each flight's scheduled time to mitigate imbalances between demand and capacity at busy airports. The second stage determines each flight's operated departure and arrival times in each operating scenario to minimize delay costs—and estimate the operating impact of each possible first-stage schedule of flights. Figure 1 illustrates these decisions.

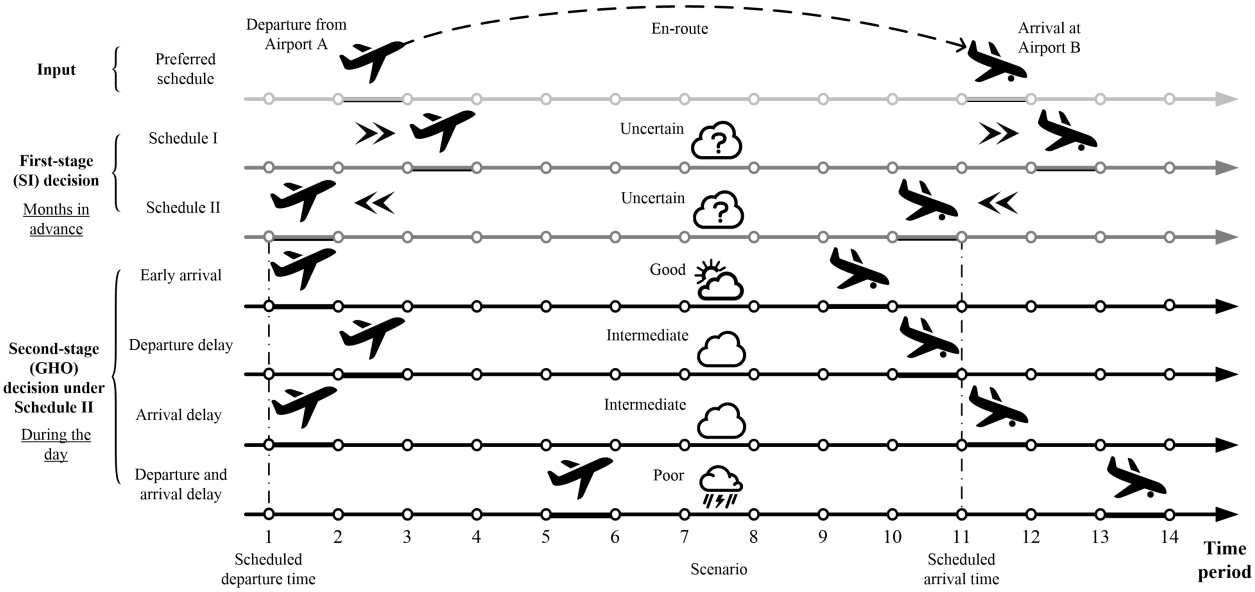



**Figure 1 An example of SI and MAGHO decisions for a hypothetical flight, under different operating scenarios.**

There are two main differences between the first-stage SI problem and the second-stage MAGHO problem. First, SI decisions face uncertainty regarding operating conditions, whereas MAGHO decisions are made once uncertainty is resolved. Second, SI can reschedule flights earlier or later than requested, whereas MAGHO can only delay departures from their scheduled times. In Figure 1, the flight requested in Periods 2–12 can be rescheduled to in Periods 1–11 or in Periods 3–13 through SI. Once scheduled in Periods 1–11, it can only depart in Period 1 or thereafter. Still, the flight can early (e.g., in Period 10) due to flexibility in en-route times.

In the first stage, SI minimizes the *schedule displacement* (i.e., deviations from airlines' preferred schedule), while maintaining all flights, block times, all aircraft connections and all passenger connections. In the second stage, MAGHO minimizes *flight delay costs*, while ensuring that en-route flying times lie within pre-specified ranges, that all aircraft connections are maintained, and that operations comply with airport capacities.

Integrating SI and MAGHO requires weighting the strategic costs of schedule displacement and the tactical costs of flight delays. Outside the United States, schedule coordination places a premium on congestion mitigation; in contrast, the unconstrained approach in place at US airports focuses

on minimizing interference with airline scheduling. The (IMSOAN) is formulated as a bi-objective model that optimizes the trade-off between schedule displacement and expected flight delays.

Note that (IMSOAN) does *not* aim to provide a unified solution to both the strategic SI and the tactical MAGHO problems—which in practice are solved by different entities at different timescales. Instead, the model aims to enhance the first-stage SI decisions by capturing their impact on downstream operating procedures and resulting delays across the network through the second-stage MAGHO problem. With the resulting schedules, operations can then be optimized by dedicated models of air traffic management and air traffic control (see, e.g., Ball et al. 2003, Balakrishnan and Chandran 2010, Solveling et al. 2011, Bertsimas et al. 2011, Jones et al. 2018).

### 3.2. Model Formulation

We formulate (IMSOAN) as a two-stage stochastic integer program. Table 1 lists notations. The network is characterized by a set $\mathscr{K}$ of airports. We discretize the day into a set $\mathscr{T}$ of 15-minute periods. For each flight $i \in \mathscr{F}$, let $S_i^{\text{dep}}$ and $S_i^{\text{arr}}$ be its requested departure and arrival times, respectively. The SI decision variables $w_{i\bullet}^{\text{dep}}$ and $w_{i\bullet}^{\text{arr}}$ are of the form $(1, \cdots, 1, 0, \cdots, 0)$, where the last "1" indicates the scheduled departure or arrival time. We apply a maximum flight displacement $\delta$—motivated by inter-flight equity concerns. Scheduling decisions are also subject to a minimum connection time $\tau_{ij}^{(1)}$ for any flight pair $(i,j) \in \mathscr{C}^1$ with an aircraft or a passenger connection.

Operating uncertainty is captured by a scenario set $\mathscr{S}$ and their probabilities $(p_s)_{s\in\mathscr{S}}$. Each scenario $s$ characterizes the operating condition at each airport $k \in \mathscr{K}$ at each time $t \in \mathscr{T}$, denoted by $\phi_{kts}$. We consider binary operating conditions, using "Visual Meteorological Conditions" (VMC) and "Instrument Meteorological Conditions" (IMC) as proxies of "good" or "poor" weather conditions, respectively. For each operating condition $\phi$, the capacity of airport $k \in \mathscr{K}$ is captured by a set of linear equations $\alpha_{kq}X + \beta_{kq}Y \leq Q_{kq}(\phi)$, for all $q \in \mathscr{Q}_k$, where $X$ and $Y$ denote the number of departures and arrivals, respectively. Parameters $\Delta_i^{\min}$ and $\Delta_i^{\max}$ specify the minimum and maximum en-route times of flight $i$. Each flight $i \in \mathscr{F}$ has a maximum departure (resp. arrival) delay $l_i^{\text{dep}}$ (resp. $l_i^{\text{arr}}$). It is motivated by equity concerns, and improves the computational efficiency of MAGHO. Finally, MAGHO is subject to a minimum connecting time $\tau_{ij}^{(2)}$ for any $(i,j) \in \mathscr{C}$ with an aircraft connection. The MAGHO variables $\boldsymbol{x}_s$ are the counterparts of the SI variables $\boldsymbol{w}$.

To reduce the number of variables in our model, we restrict the definition of the decision variables $\boldsymbol{w}$ and $\boldsymbol{x}_s$ to the relevant subset of time periods. In the first stage, each flight $i \in \mathscr{F}$ can only be scheduled to depart (resp., arrive) between period $S_i^{\text{dep}} - \delta$ and period $S_i^{\text{dep}} + \delta$ (resp., between $S_i^{\text{arr}} - \delta$ and $S_i^{\text{arr}} + \delta$). We thus only define the variables $w_{it}^{\text{dep}}$ for $t \in \tilde{\mathscr{T}}_i^{\text{dep}} = \{S_i^{\text{dep}} - \delta + 1, \cdots, S_i^{\text{dep}} + \delta\}$ and $w_{it}^{\text{arr}}$ for $t \in \tilde{\mathscr{T}}_i^{\text{arr}} = \{S_i^{\text{arr}} - \delta + 1, \cdots, S_i^{\text{arr}} + \delta\}$. By convention, $w_{it}^{\text{dep}} = 1$ for $t \leq S_i^{\text{dep}} - \delta$ and $w_{it}^{\text{dep}} = 0$ for $t > S_i^{\text{dep}} + \delta$; similarly, $w_{it}^{\text{arr}} = 1$ for $t \leq S_i^{\text{arr}} - \delta$ and $w_{it}^{\text{arr}} = 0$ for $t > S_i^{\text{arr}} + \delta$. In the

**Table 1 Notations for (IMSOAN) formulation.**

| | |
|---|---|
| **General setting:** | |
| $\mathscr{T}$ | set of time periods, $t \in \{1, \cdots, T\}$ |
| $\mathscr{F}$ | set of flights, $i, j \in \{1, \cdots, F\}$ |
| $\mathscr{K}$ | set of airports, $k \in \{1, \cdots, K\}$ |
| **The first-stage problem:** | |
| *Inputs* | |
| $\mathscr{C}^1 \subset \mathscr{F} \times \mathscr{F}$ | subset of flight pairs $(i,j) \in \mathscr{F} \times \mathscr{F}$ with an aircraft of passenger connection |
| $\tau_{ij}^{(1)}$ | minimum connecting time between flights $i$ and $j$, for all $(i,j) \in \mathscr{C}^1$ |
| $S_i^{\text{dep}}/S_i^{\text{arr}}$ | requested departure/arrival time of flight $i \in \mathscr{F}$ |
| $\delta$ | maximum displacement of all flights |
| $\tilde{\mathscr{T}}_i^{\text{dep}}/\tilde{\mathscr{T}}_i^{\text{arr}}$ | set of possible scheduled departure/arrival time periods of flight $i \in \mathscr{F}$, where $\tilde{\mathscr{T}}_i^{\text{dep}} = \{S_i^{\text{dep}} - \delta + 1, \cdots, S_i^{\text{dep}} + \delta\}$ and $\tilde{\mathscr{T}}_i^{\text{arr}} = \{S_i^{\text{arr}} - \delta + 1, \cdots, S_i^{\text{arr}} + \delta\}$ |
| $g_{it}$ | cost of displacement when flight $i$ departs in period $t$: $g_{it} = \lvert t - S_i^{\text{dep}} \rvert, \forall i \in \mathscr{F}, t \in \tilde{\mathscr{T}}_i^{\text{dep}}$ |
| *Decision variables* | |
| $w_{it}^{\text{dep}}/w_{it}^{\text{arr}}$ | 1 if flight $i \in \mathscr{F}$ is scheduled to depart/arrive no earlier than period $t \in \tilde{\mathscr{T}}_i^{\text{dep}}$/ $t \in \tilde{\mathscr{T}}_i^{\text{arr}}$, 0 otherwise |
| **The second-stage problem:** | |
| *Inputs* | |
| $\mathscr{S}$ | set of operating scenarios, $s \in \{1, \cdots, S\}$ |
| $\mathscr{F}_k^{\text{dep}}/\mathscr{F}_k^{\text{arr}}$ | set of departing/arriving flights at airport $k \in \mathscr{K}$ |
| $\mathscr{Q}_k$ | set of capacity constraints at airport $k \in \mathscr{K}$ |
| $\mathscr{C}^2 \subset \mathscr{F} \times \mathscr{F}$ | subset of flight pairs $(i,j) \in \mathscr{F} \times \mathscr{F}$ with an aircraft connection |
| $\tau_{ij}^{(2)}$ | minimum connecting time between flights $i$ and $j$, for all $(i,j) \in \mathscr{C}^2$ |
| $\Phi_{kt}$ | random operating condition at airport $k \in \mathscr{K}$ at time $t \in \mathscr{T}$ |
| $\phi_{kts}$ | operating condition realization of $\Phi_{kt}$, at airport $k \in \mathscr{K}$ at time $t \in \mathscr{T}$ in scenario $s \in \mathscr{S}$ |
| $p_s$ | probability of occurrence of scenario $s \in \mathscr{S}$ |
| $a_{kq}, b_{kq}, Q_{kq}(\phi)$ | parameters of capacity envelope at airport $k \in \mathscr{K}$ under operating condition $\phi$, for $q \in \mathscr{Q}_k$ |
| $\Delta_i^{\min}/\Delta_i^{\max}/\Delta_i^{\text{sch}}$ | minimum/maximum/scheduled en-route time of flight $i \in \mathscr{F}$, with $\Delta_i^{\min} \le \Delta_i^{\text{sch}} \le \Delta_i^{\max}$ and $\Delta_i^{\text{sch}} = S_i^{\text{arr}} - S_i^{\text{dep}}$ |
| $l_i^{\text{dep}}/l_i^{\text{arr}}$ | maximum departure/arrival delay allowed for flight $i \in \mathscr{F}$ |
| $\mathscr{T}_i^{\text{dep}}/\mathscr{T}_i^{\text{arr}}$ | set of possible operating departure/arrival time periods of flight $i \in \mathscr{F}$, where $\mathscr{T}_i^{\text{dep}} = \{S_i^{\text{dep}} - \delta + 1, \cdots, S_i^{\text{dep}} + \delta + l_i^{\text{dep}}\}$ and $\mathscr{T}_i^{\text{arr}} = \{S_i^{\text{dep}} - \delta + \Delta_i^{\min} + 1, \cdots, S_i^{\text{arr}} + \delta + l_i^{\text{arr}}\}$ |
| $c_i^{\text{dep}}/c_i^{\text{arr}}$ | unit cost of departure/arrival delay per period of flight $i \in \mathscr{F}$, with $c_i^{\text{dep}} \le c_i^{\text{arr}}$ |
| *Decision variables* | |
| $x_{its}^{\text{dep}}/x_{its}^{\text{arr}}$ | 1 if flight $i \in \mathscr{F}$ departs/arrives no earlier than period $t \in \mathscr{T}_i^{\text{dep}}/\mathscr{T}_i^{\text{arr}}$ in scenario $s \in \mathscr{S}$, 0 otherwise |
| $v_{is}^{\text{dep}}/v_{is}^{\text{arr}}$ | departure/arrival delay of flight $i \in \mathscr{F}$ in scenario $s \in \mathscr{S}$ |

We use bold symbols to refer to vectors, e.g., $\boldsymbol{x}_s^{\text{dep}} = \{x_{its}^{\text{dep}} |\ i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{dep}}\}$ and $\boldsymbol{x}_s = (\boldsymbol{x}_s^{\text{dep}}, \boldsymbol{x}_s^{\text{arr}})$.

second stage, flight $i \in \mathscr{F}$ cannot depart before period $S_i^{\text{dep}} - \delta$ or after period $S_i^{\text{dep}} + \delta + l_i^{\text{dep}}$; and it cannot arrive before period $S_i^{\text{dep}} - \delta + \Delta_i^{\min}$ or after period $S_i^{\text{arr}} + \delta + l_i^{\text{arr}}$. Therefore, we only define the variables $x_{its}^{\text{dep}}$ for $t \in \mathscr{T}_i^{\text{dep}} = \{S_i^{\text{dep}} - \delta + 1, \cdots, S_i^{\text{dep}} + \delta + l_i^{\text{dep}}\}$ and $x_{its}^{\text{arr}}$ for $t \in \mathscr{T}_i^{\text{arr}} =$

$\{S_i^{\text{dep}} - \delta + \Delta_i^{\min} + 1, \cdots, S_i^{\text{arr}} + \delta + l_i^{\text{arr}}\}$. Again, by convention, $x_{its}^{\text{dep}} = 1$ for $t \leq S_i^{\text{dep}} - \delta$ and $x_{its}^{\text{dep}} = 0$ for $t > S_i^{\text{dep}} + \delta + l_i^{\text{dep}}$; similarly, $x_{its}^{\text{arr}} = 1$ for $t \leq S_i^{\text{dep}} - \delta + \Delta_i^{\min}$ and $x_{its}^{\text{arr}} = 0$ for $t > S_i^{\text{arr}} + \delta + l_i^{\text{arr}}$.

Using notations from Table 1 and the above specifications, (IMSOAN) is formulated as follows:

$$\text{(IMSOAN)} \qquad \min \rho \sum_{i \in \mathscr{F}} \sum_{t \in \tilde{\mathscr{T}}_i^{\text{dep}}} g_{it}(w_{i,t-1}^{\text{dep}} - w_{it}^{\text{dep}}) + (1-\rho)\mathbb{E}_{\boldsymbol{\Phi}}[\Psi(\boldsymbol{w})] \tag{1a}$$

$$\text{s.t. } w_{it}^{\text{dep}} \leq w_{i,t-1}^{\text{dep}} \quad \forall i \in \mathscr{F},\ t \in \tilde{\mathscr{T}}_i^{\text{dep}}, \tag{1b}$$

$$w_{it}^{\text{arr}} \leq w_{i,t-1}^{\text{arr}} \quad \forall i \in \mathscr{F},\ t \in \tilde{\mathscr{T}}_i^{\text{arr}}, \tag{1c}$$

$$\sum_{t \in \mathscr{T}} (w_{it}^{\text{arr}} - w_{it}^{\text{dep}}) = \Delta_i^{\text{sch}}, \quad \forall i \in \mathscr{F}, \tag{1d}$$

$$\sum_{t \in \mathscr{T}} (w_{jt}^{\text{dep}} - w_{it}^{\text{arr}}) \geq \tau_{ij}^{(1)} \quad \forall (i,j) \in \mathscr{C}^1, \tag{1e}$$

$$w_{it}^{\text{dep}},\ w_{it'}^{\text{arr}} \in \{0,1\} \quad \forall i \in \mathscr{F},\ t \in \tilde{\mathscr{T}}_i^{\text{dep}},\ t' \in \tilde{\mathscr{T}}_i^{\text{arr}}. \tag{1f}$$

In (IMSOAN), $\mathbb{E}_{\boldsymbol{\Phi}}[\Psi(\boldsymbol{w})]$ evaluates the expected delay cost from the second-stage MAGHO problem over all scenarios $\boldsymbol{\Phi}$, given any SI solution $\boldsymbol{w}$. Equation (1a) minimizes the schedule displacement (weighted by parameter $\rho$) and expected delay costs (weighted by $1-\rho$). Constraints (1b) and (1c) guarantee that variables $\boldsymbol{w}$ are non-increasing in $t$. Constraint (1d) maintains the scheduled en-route times of all flights. Constraint (1e) applies the minimum connecting times between relevant flight pairs. Constraint (1f) defines the domain of all variables.

The expected second-stage cost $\mathbb{E}_{\boldsymbol{\Phi}}[\Psi(\boldsymbol{w})]$ is given by $\sum_{s \in \mathscr{S}} p_s \Psi(\boldsymbol{w}, \boldsymbol{\phi}_s)$, where $\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s)$ denotes the MAGHO cost in operating condition $\boldsymbol{\phi}_s$ in scenario $s \in \mathscr{S}$. It is obtained as follows:

$$\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s) = \min \sum_{i \in \mathscr{F}} (c_i^{\text{dep}} v_{is}^{\text{dep}} + c_i^{\text{arr}} v_{is}^{\text{arr}}), \tag{2a}$$

$$\text{s.t. } x_{its}^{\text{dep}} \leq x_{i,t-1,s}^{\text{dep}} \quad \forall i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{dep}}, \tag{2b}$$

$$x_{its}^{\text{arr}} \leq x_{i,t-1,s}^{\text{arr}} \quad \forall i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{arr}}, \tag{2c}$$

$$\sum_{t \in \mathscr{T}} (x_{its}^{\text{dep}} - w_{it}^{\text{dep}}) = v_{is}^{\text{dep}} \quad \forall i \in \mathscr{F}, \tag{2d}$$

$$\sum_{t \in \mathscr{T}} (x_{its}^{\text{arr}} - w_{it}^{\text{arr}}) \leq v_{is}^{\text{arr}} \quad \forall i \in \mathscr{F}, \tag{2e}$$

$$v_{is}^{\text{dep}} \leq l_i^{\text{dep}} \quad \forall i \in \mathscr{F}, \tag{2f}$$

$$v_{is}^{\text{arr}} \leq l_i^{\text{arr}} \quad \forall i \in \mathscr{F}, \tag{2g}$$

$$\sum_{t \in \mathscr{T}} (x_{jts}^{\text{dep}} - x_{its}^{\text{arr}}) \geq \tau_{ij}^{(2)} \quad \forall (i,j) \in \mathscr{C}^2, \tag{2h}$$

$$\sum_{t \in \mathscr{T}} (x_{its}^{\text{arr}} - x_{its}^{\text{dep}}) \geq \Delta_i^{\min} \quad \forall i \in \mathscr{F}, \tag{2i}$$

$$\sum_{t \in \mathscr{T}} (x_{its}^{\text{arr}} - x_{its}^{\text{dep}}) \leq \Delta_i^{\max} \quad \forall i \in \mathscr{F}, \tag{2j}$$

$$a_{kq} \sum_{i \in \mathscr{F}_k^{\text{dep}}} (x_{i,t-1,s}^{\text{dep}} - x_{its}^{\text{dep}}) + b_{kq} \sum_{i \in \mathscr{F}_k^{\text{arr}}} (x_{i,t-1,s}^{\text{arr}} - x_{its}^{\text{arr}}) \leq Q_{kq}(\phi_{kts})$$
$$\forall q \in \mathscr{Q}_k,\ t \in \mathscr{T},\ k \in \mathscr{K}, \quad \text{(2k)}$$

$$x_{its}^{\text{dep}} - w_{it}^{\text{dep}} \geq 0 \quad \forall i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{dep}}, \quad \text{(2l)}$$

$$x_{i,t-\left(\Delta_i^{\text{sch}} - \Delta_i^{\min}\right),s}^{\text{arr}} - w_{it}^{\text{arr}} \geq 0 \quad \forall i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{arr}}, \quad \text{(2m)}$$

$$x_{its}^{\text{dep}}, x_{it's}^{\text{arr}} \in \{0,1\} \quad \forall i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{dep}},\ t' \in \mathscr{T}_i^{\text{arr}}, \quad \text{(2n)}$$

$$v_{is}^{\text{dep}}, v_{is}^{\text{arr}} \geq 0 \quad \forall i \in \mathscr{F}. \quad \text{(2o)}$$

Equation (2a) minimizes departure and arrival delay costs. Constraints (2b) and (2c) ensure that variables $\boldsymbol{x}_s$ are non-increasing in $t$. Constraint (2d) defines the departure delay $v_{is}^{\text{dep}}$ as the difference between flight $i$'s departure time in scenario $s$ and its scheduled departure time. Since $\boldsymbol{v}^{\text{dep}} \geq 0$ (Constraint (2o)), Constraint (2d) also enforces that flight $i$ cannot depart earlier than scheduled. Constraint (2e) defines the arrival delay as the difference between flight $i$'s arrival time in scenario $s$ and its scheduled arrival time, if positive (and zero otherwise). Indeed, since $\boldsymbol{v}^{\text{arr}} \geq 0$ (Constraint (2o)) and given the model's delay-minimization objective (Equation (2a)), Constraint (2e) ensures that $v_{is}^{\text{arr}} = \max\{0, \sum_{t \in \mathscr{T}}(x_{its}^{\text{arr}} - w_{it}^{\text{arr}})\}$ for all flights $i \in \mathscr{F}$ in all scenarios $s \in \mathscr{S}$. Note that Constraint (2d) allows each flight $i$ to arrive earlier than scheduled. Constraints (2f) and (2g) apply the maximum delay for each flight. Constraint (2h) applies the minimum connecting time between any flight pair $(i,j) \in \mathscr{C}^2$. Constraints (2i) and (2j) ensure that the en-route time of each flight lies within the acceptable range. Constraints (2k) guarantee that the number of flights operated at time $t \in \mathscr{T}$ at airport $k \in \mathscr{K}$ does not exceed its realized capacity.

Next, Constraints (2l) and (2m) are valid inequalities that tighten the linear programming relaxation of (IMSOAN)—leveraging the fact that each flight $i \in \mathscr{F}$ cannot depart earlier than scheduled and can save up to $\Delta_i^{\text{sch}} - \Delta_i^{\min}$ en-route. We prove these valid inequalities formally in Appendix A. Constraints (2l) and (2m) also yield lower bounds of the second-stage variables $\boldsymbol{x}_s$ (i.e., when flights are operated) based on the first-stage variables $\boldsymbol{w}$ (i.e., when flights are scheduled). This relationship will be central to our solution approach in Section 4.

### 3.3. Size and Complexity

(IMSOAN) can be viewed as an integer program with constraints (1b)–(1f) applied across all scenarios, and scenario-dependent constraints (2b)–(2o). The first-stage problem has $4 \cdot F \cdot \delta$ variables; the second-stage problem has $\sum_{i \in \mathscr{F}} \left(|\mathscr{T}_i^{\text{dep}}| + |\mathscr{T}_i^{\text{arr}}|\right) + F = 4 \cdot F \cdot \delta + \sum_{i \in \mathscr{F}}(l_i^{\text{dep}} + l_i^{\text{arr}} + \Delta_i^{\text{sch}} - \Delta_i^{\min})$ variables in each scenario $s \in \mathscr{S}$.[2] The first-stage problem has $(4\delta + 1) \cdot F + |\mathscr{C}^1|$ constraints; the second-stage problem has $2 \cdot \sum_{i \in \mathscr{F}} \left(|\mathscr{T}_i^{\text{dep}}| + |\mathscr{T}_i^{\text{arr}}|\right) + 6F + |\mathscr{C}^2| + T \cdot \sum_{k \in \mathscr{K}} |\mathscr{Q}_k|$ constraints in each

[2] This expression excludes variables $v_{is}^{\text{dep}}$ since they can be derived from $\sum_{t \in \mathscr{T}}(x_{its}^{\text{dep}} - w_{it}^{\text{dep}})$ (Constraint (2d)).

scenario $s \in \mathscr{S}$. In our largest test instance, (IMSOAN) involves over 8.2 million integer variables and 20.8 million constraints (see Table 2). It is thus highly intractable with commercial solvers.

Viewed as a two-stage stochastic program, the main challenge in solving (IMSOAN) lies in approximating the expected second-stage cost $\mathbb{E}_{\boldsymbol{\Phi}}[\Psi(\boldsymbol{w})]$, which is non-convex and non-continuous. Moreover, the structure of (IMSOAN) makes it particularly intractable. First, the second-stage MAGHO problem is NP-hard, which complicates the evaluation of the recourse function. Second, the first-stage problem admits an exponentially large solution space, with up to $(2 \cdot \delta + 1)^F$ feasible solutions (in the absence of connectivity constraints (1e)). Thus, (IMSOAN) is not amenable to being solved with the integer L-shaped method, which is most effective when the first-stage problem has a limited feasible region (Laporte and Louveaux 1993). Due to the complexity of (IMSOAN), we design next an original algorithm to solve it efficiently.

## 4. A Two-stage Integer Programming Solution Approach

We provide an overview of our two-stage stochastic integer programming solution approach in Section 4.1 and develop it in Sections 4.2–4.5. Section 4.6 details our solution algorithm for (IMSOAN) and its generalizability to other problems. All proofs in this section are given in Appendix B.

Before proceeding, we write a compact formulation of (IMSOAN) by decomposing it into a first-stage master problem (MP) and a second-stage sub-problem (SP) (that capture our SI problem and MAGHO problem, respectively). The sub-problem (SP) is formulated in standard form below. The right-hand side depends on the first-stage solution $\boldsymbol{w}$ and on the operating conditions $\boldsymbol{\Phi}$ in scenario $s \in \mathscr{S}$, and is denoted by $\boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s)$. Variables $\boldsymbol{y}_s \geq \boldsymbol{0}$ denote slack variables. Note that matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ do not depend on the scenario $s$ (see Equations (2b)–(2o)). We refer to the linear programming (LP) relaxation of (SP) as $\left(\overline{\text{SP}}\right)$ and to its dual problem as $\left(\text{D} - \overline{\text{SP}}\right)$.

$$\text{(SP)} \qquad \Psi(\boldsymbol{w}, \boldsymbol{\phi}_s) = \min \{\boldsymbol{c}\boldsymbol{v}_s \mid \boldsymbol{A}_1\boldsymbol{x}_s + \boldsymbol{A}_2\boldsymbol{v}_s + \boldsymbol{E}\boldsymbol{y}_s = \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s), \boldsymbol{x}_s \text{ binary}, \ \boldsymbol{v}_s \geq 0, \ \boldsymbol{y}_s \geq 0.\} \quad (3)$$

We formulate the master problem (MP) as follows, where $\boldsymbol{W}$ denotes the feasible set of variables $\boldsymbol{w}$ that satisfy Equations (1b)–(1f) and $\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s)$ is obtained by Equation (3). The main difficulty stems from approximating the recourse function (Equation (4b)).

$$\text{(MP)} \qquad \min \ \rho \sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\text{dep}}} g_{it}(w_{i,t-1}^{\text{dep}} - w_{it}^{\text{dep}}) + (1-\rho) \sum_{s \in \mathscr{S}} p_s \theta_s, \quad (4a)$$

$$\text{s.t.} \ \ \theta_s \geq \Psi(\boldsymbol{w}, \boldsymbol{\phi}_s), \quad \forall s \in \mathscr{S}, \quad (4b)$$

$$\boldsymbol{w} \in \boldsymbol{W}, \ \boldsymbol{\theta} \geq 0. \quad (4c)$$

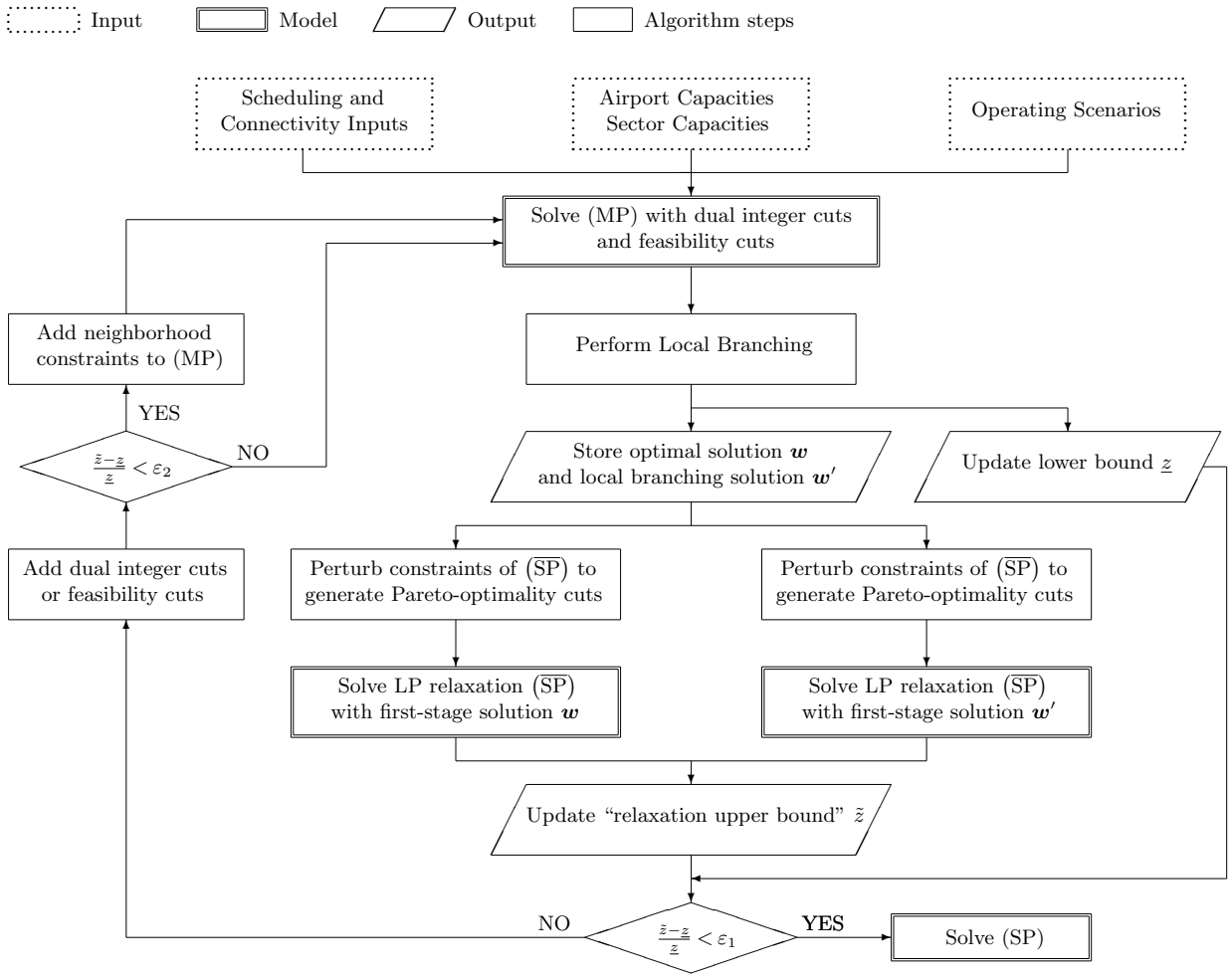



**Figure 2 Overview of the proposed solution architecture for two-stage stochastic integer programming.**

### 4.1. Overview of Solution Approach

Figure 2 provides an overview of our two-stage stochastic integer programming solution architecture, which decomposes the problem by iterating between the (MP) and (SP) in each scenario.

The central component of our solution framework is a new set of optimality cuts (Section 4.2), used to approximate the recourse function (Equation (4b)) with linear equations based on first-stage variables. The optimality cuts are based on the dual variables and reduced costs of $\left(\text{D}-\overline{\text{SP}}\right)$—the dual LP relaxation of (SP), and thus referred to as *dual integer cuts*. At each iteration, the algorithm solves (MP) with the optimality and feasibility cuts from previous iterations. This provides a feasible first-stage solution $\boldsymbol{w}$ and a corresponding lower bound of (IMSOAN), denoted by $\underline{z}$. We then solve $\left(\overline{\text{SP}}\right)$ in each scenario $s \in \mathscr{S}$. If it admits a feasible solution, we add new optimality cuts to (MP); otherwise, we add new feasibility cuts (Section 4.3). We update a variable $\tilde{z}$, named "relaxation upper bound", as follows: $\tilde{z} \leftarrow \min\left\{\tilde{z},\ \rho\sum_{i\in\mathscr{F}}\sum_{t\in\tilde{\mathscr{T}}_i^{\text{dep}}} g_{it}(w_{i,t-1}^{\text{dep}} - w_{it}^{\text{dep}}) + (1-\rho)\sum_{s\in\mathscr{S}} p_s \bar{\Psi}(\boldsymbol{w},\boldsymbol{\phi}_s)\right\}$, where $\bar{\Psi}(\boldsymbol{w},\boldsymbol{\phi}_s)$

denotes the value of the objective function of $\left(\overline{\text{SP}}\right)$ in scenario $s$ given (MP) solution $\boldsymbol{w}$. Note that $\tilde{z}$ is *not* necessarily an upper bound of (IMSOAN), but is used to control the algorithm's convergence: the procedure terminates when the gap between $\underline{z}$ and $\tilde{z}$ lies within $\varepsilon_1$ (e.g., 1%). Upon termination, we store the (MP) solution, denoted by $\boldsymbol{w}^*$; we then solve (SP) in each scenario. If it is infeasible, we add integer feasibility cuts; otherwise, we denote the solution by $\overline{z}$, which is a valid upper bound of (IMSOAN). Ultimately, the algorithm provides provable solution guarantees, characterized by the gap between $\underline{z}$ and $\overline{z}$.

Once the convergence gap between $\underline{z}$ and $\tilde{z}$ reaches a looser threshold $\varepsilon_2$ (e.g., 5%), we add original *neighborhood constraints* to (MP) to overcome a long-tail effect in the procedure's convergence (Section 4.4). These constraints restrict the first-stage solutions to the neighborhoods of solutions generated in initial iterations (before the gap reaches $\varepsilon_2$)—effecting a shift from exploration to exploitation. Note that this approach does not provide approximate solutions and is thus distinct from neighborhood search heuristics. Instead, it orients the first-stage search toward the most promising regions to accelerate (MP) and strengthens the dual integer cuts and feasibility cuts.

Finally, since our solution approach solves the linear programming relaxation of our second-stage problem, we can leverage two standard stochastic programming acceleration techniques: local branching and Pareto-optimality cuts (Section 4.5 and Appendix B.6).

### 4.2. Optimality Cuts: Dual Integer Cuts

The main element of our solution approach is the *dual integer cut*—an optimality cut in the first-stage (MP) used to approximate the recourse function (Equation (4b)). This cut is valid with any feasible solutions of (MP) and $\left(\text{D}-\overline{\text{SP}}\right)$. In our algorithm, we generate it from the optimal solution of $\left(\text{D}-\overline{\text{SP}}\right)$, using feasible (MP) solutions obtained at each iteration.

Let $\boldsymbol{\Pi}_s$ denote a feasible solution of $\left(\text{D}-\overline{\text{SP}}\right)$, the dual LP relaxation of (SP) (formulated in Appendix B.1). Proposition 1 expresses the optimal value of the objective function of (SP) as the objective value realized by $\boldsymbol{\Pi}_s$, plus the sum, over all primal variables, of their optimal integer values in (SP) and their reduced costs from $\left(\overline{\text{SP}}\right)$.

PROPOSITION 1. *Let $\boldsymbol{w}$ be a feasible solution of (MP), and $\boldsymbol{\Pi}_s$ be a feasible solution of $\left(D-\overline{SP}\right)$ in scenario $s\in\mathscr{S}$. Let also $(\boldsymbol{x}_s^*,\boldsymbol{v}_s^*,\boldsymbol{y}_s^*)$ denote the optimal solutions of (SP) in scenario $s\in\mathscr{S}$ with respect to $\boldsymbol{w}$. Let $\gamma'(\boldsymbol{x}_s),\gamma'(\boldsymbol{v}_s),\gamma'(\boldsymbol{y}_s)\geq 0$ be the reduced cost of the primal variables of $\left(\overline{SP}\right)$ with respect to $\boldsymbol{\Pi}_s$ in scenario $s$. The optimal value of (SP) in scenario $s$ satisfies:*

$$\Psi(\boldsymbol{w},\boldsymbol{\phi}_s)=\boldsymbol{\Pi}_s\boldsymbol{b}(\boldsymbol{w},\boldsymbol{\phi}_s)+\boldsymbol{x}_s^*\gamma'(\boldsymbol{x}_s)+\boldsymbol{v}_s^*\gamma'(\boldsymbol{v}_s)+\boldsymbol{y}_s^*\gamma'(\boldsymbol{y}_s). \tag{5}$$

Equation (5) does not provide yet a useful constraint into (MP) because it relies on the (unknown) second-stage solutions $(\boldsymbol{x}_s^*,\boldsymbol{v}_s^*,\boldsymbol{y}_s^*)$. Nonetheless, we derive in Theorem 1 a lower bound for $\Psi(\boldsymbol{w},\boldsymbol{\phi}_s)$

based on first-stage variables alone. This lower bound provides a valid optimality cut—our *dual integer cut*—to approximate Equation (4b) in (MP). This is obtained from valid inequalities (2l) and (2m), which bound the second-stage variable $\boldsymbol{x}_s^*$ as a function of the first-stage variable $\boldsymbol{w}$. The last two terms in Equation (5) are simply bounded below by zero. This still yields a tight bound, because the main primal and dual information required to evaluate $\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s)$ is stored in the first two terms of Equation (5)—see Appendix B.3 for details on this point.

THEOREM 1. *Let $\boldsymbol{w}$ be a feasible solution of (MP), and $\boldsymbol{\Pi}_s$ be a feasible solution of $\left(D-\overline{SP}\right)$ in scenario $s \in \mathscr{S}$. Let $\gamma'(\boldsymbol{x}_s), \gamma'(\boldsymbol{v}_s), \gamma'(\boldsymbol{y}_s) \geq 0$ be the reduced cost of the primal variables of $\left(\overline{SP}\right)$ with respect to $\boldsymbol{\Pi}_s$ in scenario $s$. Equation (6) is a valid cut for (MP):*

$$\theta_s \geq \boldsymbol{\Pi}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi_s}) + \sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\text{dep}}} w_{it}^{\text{dep}} \gamma'\left(x_{its}^{\text{dep}}\right) + \sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\text{arr}}} w_{it}^{\text{arr}} \gamma'\left(x_{i,t-\left(\Delta_{in}^{\text{sch}}-\Delta_{in}^{\min}\right),s}^{\text{arr}}\right), \tag{6}$$

*where, by convention, $\gamma'\left(x_{it's}^{\text{arr}}\right) = 0$ for all $t' \notin \mathscr{T}_i^{\text{arr}}$ and for all $i \in \mathscr{F}$.*

Equation (6) provides tight optimality cuts, i.e., a tighter lower bound for $\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s)$. To see this, note that traditional Benders optimality cut—valid for any two-stage stochastic integer program—is of the form $\theta_s \geq \boldsymbol{\Pi}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi_s}), \forall s \in \mathscr{S}$. Our dual integer cuts incorporates the additional (non-negative) terms $\sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\text{dep}}} w_{it}^{\text{dep}} \gamma'\left(x_{its}^{\text{dep}}\right)$ and $\sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\text{arr}}} w_{it}^{\text{arr}} \gamma'\left(x_{i,t-\left(\Delta_{in}^{\text{sch}}-\Delta_{in}^{\min}\right),s}^{\text{arr}}\right)$. Stated differently, our dual integer cut tightens traditional Benders cuts by leveraging the reduced cost of the dual linear programming relaxation of the second-stage problem and the structure of (IM-SOAN) (i.e., no early departure allowed in MAGHO). Our results will show that these added terms significant improve the performance of our solution algorithm, as compared to Benders cuts.

Our optimality cuts leverage dual and residual information of the second-stage problem (see Proposition 1 and Lemma 1). As mentioned in Section 2.2, this approach shares similarities with the additive bounding procedure from Fischetti and Toth (1989). However, the additive bounding procedure uses this expression to derive a lower bound for combinatorial optimization problems. In contrast, we use it, along with a relationship between first-stage and second-stage variables, to derive our dual integer cut for two-stage stochastic integer programming.

### 4.3. Feasibility Cuts

A feasibility cut is required in two instances. First, at each iteration of the algorithm, a feasible (MP) solution $\boldsymbol{w}$ may lead to an empty feasible region for $\left(\overline{\text{SP}}\right)$. The traditional Benders feasibility cut can be directly applied here to the continuous relaxation of (SP). This cut is formulated in Appendix B.5. Second, upon completion of the iterative algorithm, the final (MP) solution $\boldsymbol{w}^*$ leads to a non-empty feasible region for $\left(\overline{\text{SP}}\right)$ but may not lead to a non-empty feasible region for (SP). We thus add an integer feasibility cut to (MP) in Proposition 2, following the integer L-shaped principles from Laporte and Louveaux (1993). We repeat the process until a feasible integer solution is found.

PROPOSITION 2. *Let $\widehat{\boldsymbol{w}}$ denote a feasible integer solution of (MP) such that the corresponding (SP) is infeasible in scenario $s \in \mathscr{S}$. Equation* (7) *is a valid integer feasibility cut in (MP).*

$$\sum_{i\in\mathscr{F}}\sum_{t\in\mathscr{T}_i^{\text{dep}}} \widehat{w}_{it}^{\text{dep}} w_{it}^{\text{dep}} - \sum_{i\in\mathscr{F}}\sum_{t\in\mathscr{T}_i^{\text{dep}}} (1-\widehat{w}_{it}^{\text{dep}}) w_{it}^{\text{dep}} \leq \sum_{i\in\mathscr{F}}\sum_{t\in\mathscr{T}_i^{\text{dep}}} \widehat{w}_{it}^{\text{dep}} - 1. \tag{7}$$

Computationally, we never encountered instances where (SP) is integer infeasible for the final (MP) solution when $\left(\overline{\text{SP}}\right)$ admits a feasible solution.[3] Thus, Equation (7) was never added to (MP).

### 4.4. Acceleration Based on Neighborhood Constraints

The iterative algorithm exhibits a long-tail effect, i.e., improvements in solution quality become increasingly smaller. To overcome this effect, we restrict the first-stage (MP) solutions in later iterations to the neighborhood of those generated in earlier iterations. This is achieved by adding *neighborhood constraints* to (MP) once the convergence gap reaches a pre-determined threshold. This approach speeds up the algorithm without sacrificing optimality or solution quality guarantees.

Let $\boldsymbol{w}^{(r)}$ be the variables $\boldsymbol{w}$ obtained from solving (MP) at the $r^{\text{th}}$ iteration of the algorithm. We define a set $\Omega^r \subset \mathscr{F}$ that includes all flights that have been displaced up to iteration $r$, i.e., $\Omega^r = \cup_{\upsilon=1,\cdots,r}\{i\in\mathscr{F}|\ \sum_{t\in\mathscr{T}} w_{it}^{\text{dep}(\upsilon)} \neq S_i^{\text{dep}}\}$. After several iterations, $|\Omega^{\widehat{r}}| - |\Omega^{\widehat{r}-1}|$ becomes marginal, i.e., almost all flights displaced in (MP) were already displaced in earlier iterations. We thus define a threshold $\varepsilon_2$, and denote by $\widehat{r}$ the first iteration such that $\frac{\tilde{z}-\underline{z}}{\underline{z}} \leq \varepsilon_2$. From iteration $\widehat{r}+1$ onward, we add Equation (8)—a neighborhood constraint—to (MP), which ensures that all flights that have not been displaced up to iteration $\widehat{r}$ will not be displaced either in later iterations. This constraint restricts the feasible solution space of (MP) but does not affect the validity of the dual integer cuts (Equation (6)) and feasibility cuts (Equation (7))—which hold for *any* feasible (MP) solution.

$$\sum_{t\in\mathscr{T}} w_{it}^{\text{dep}} - S_i^{\text{dep}} = 0 \quad \forall i \in \mathscr{F} \setminus \Omega^{\widehat{r}}. \tag{8}$$

When Constraint (8) is applied, the objective value of (MP), denoted as $\underline{z}'$, does not necessarily provide a global lower bound to (IMSOAN). Upon termination, i.e., when the gap $\frac{\tilde{z}-\underline{z}'}{\underline{z}'}$ lies within $\varepsilon_1$, we solve (MP) without the neighborhood constraints (Equation (8)) but with optimality and feasibility cuts generated in *all* iterations (including those with the neighborhood constraints). This updates the lower bound $\underline{z}$. By construction, $\underline{z} \leq \underline{z}'$, but we found computationally that $\underline{z} \approx \underline{z}'$ as long as constraint (8) is added at an appropriate threshold $\varepsilon_2$ (when the long-tail effect occurs).

The neighborhood constraints provide three benefits. First, they significantly restrict the solution space of (MP), thus enhancing its computational efficiency. Second, this restriction of the feasible

[3] This can be explained by the structure of (SP). Equations (2b) and (2c) provide facet-defining constraints (Bertsimas and Stock Patterson 1998) and Equations (2d) to (2j) define a totally unimodular matrix. (SP) could still be infeasible due to capacity constraints (Equations (2k)) but we have not encountered this instance.

space of (MP) results in faster updates of the relaxation upper bound $\tilde{z}$. Third, the resulting optimality cuts are more tailored to flights in the subset $\Omega^{\hat{r}} \subset \mathscr{F}$, which are the most "promising" ones, leading to stronger improvements of the lower bound $\underline{z}$ in final iterations.

### 4.5. Acceleration Based on Continuous Subproblem Techniques

Recall that our solution algorithm relies on the linear programming relaxation of the sub-problem. We can thus leverage two acceleration techniques from stochastic continuous programming: local branching (Fischetti and Lodi 2003, Rei et al. 2009) and Pareto-optimality cuts (Magnanti and Wong 1981, Sherali and Lunday 2013). First, we solve (MP) at each iteration using local branching, which yields multiple feasible (MP) solutions. We extract the optimal solution $\boldsymbol{w}$ and the second-best solution $\boldsymbol{w}'$, and thus derive two sets of optimality or feasibility cuts in parallel. Second, when solving $\left(\overline{\text{SP}}\right)$ with (MP) solution $\boldsymbol{w}$, we perturb the constraints of $\left(\overline{\text{SP}}\right)$ by considering a "core point" $\boldsymbol{w}^{(o)}$ of the feasible region of (MP). This yields non-dominated cuts in $\boldsymbol{w}$ and $\boldsymbol{w}^{(o)}$. Since these techniques are well established, we defer implementation details to Appendix B.6.

### 4.6. Full Algorithm and Generalizability

Algorithm 1 synthesizes our stochastic integer programming solution approach. The procedure terminates when the convergence gap reaches $\varepsilon_1$ or after $r^*$ iterations, whichever comes first. We then solve (MP) to get solution $\boldsymbol{w}^*$, and solve (SP) to derive solution $\boldsymbol{x}_s^*$ in each scenario $s \in \mathscr{S}$.

The dual integer cuts and neighborhood constraints are generalizable to similarly structured problems. First, the dual integer cut can solve any two-stage stochastic program with integer recourse, as long as the second-stage variables can be bounded below by linear terms involving first-stage variables. Indeed, the cut expresses the recourse function with reduced costs and second-stage variables (Proposition 1), which can be used to derive a global lower bound of the recourse function (Theorem 1). Of course, this method is more effective when the linear combination of first-stage variables provides a tighter lower bound of second-stage variables.

Second, the dual integer cut is also valid for two-stage stochastic programs with continuous recourse. Following our proof of Proposition 1, any continuous recourse function can be expressed in a form similar to Equation (5). Thus, one can build a global optimality cut that dominates traditional Benders cuts, as in Equation (6)—again, when second-stage variables can be bounded below by a linear combination of first-stage variables. Ultimately, this approach can be generalized to a broad class of problems involving strategic decisions under uncertainty (e.g., network design, scheduling, pricing) and tactical decisions made once uncertainty is resolved (e.g., operations).

Third, our neighborhood constraints can be used to accelerate the procedure for two-stage stochastic programs that optimize a small set of variables over a large set of first-stage variables (e.g., selecting a small set of facilities over many candidate locations). These constraints shift the

**Algorithm 1** Two-stage stochastic integer programming algorithm for (IMSOAN).

**Initialization.** Initialize parameters and sets.

(a) Set iteration $r=0$, maximum iteration $r^*=70$, lower bounds $\underline{z}=\underline{z}'=10^{-6}$, relaxation upper bound $\tilde{z}=+\infty$, upper bound $\overline{z}=+\infty$, convergence thresholds $\varepsilon_1=0.01, \varepsilon_2=0.05$.

(b) Initialize sets of dual integer cuts $\Theta^D=\emptyset$, feasibility cuts $\Theta^F=\emptyset$, displaced flights $\Omega=\emptyset$.

(c) Solve LP relaxation of (IMSOAN) to derive the core point $\boldsymbol{w}^{(o)}$.

**Iterate between Step 1 and Step 2, defined below, until convergence**

**Step 1.** Update $r \leftarrow r+1$; Solve (MP):

(a) Add cuts from $\Theta^D$ and $\Theta^F$; If $\frac{\tilde{z}-\underline{z}}{\underline{z}} < \varepsilon_2$, add neighborhood constraints (8).

(b) Solve (MP) with local branching; get optimal solution $\boldsymbol{w}$ and second-best solution $\boldsymbol{w}'$.

(c) Update $\Omega \leftarrow \Omega \cup \{i \in \mathscr{F} |\ \sum_{t\in\mathscr{T}} w_{it}^{\text{dep}} - S_i^{\text{dep}} \neq 0\} \cup \{i \in \mathscr{F} |\ \sum_{t\in\mathscr{T}} w_{it}^{\text{dep}'} - S_i^{\text{dep}} \neq 0\}$; If $\frac{\tilde{z}-\underline{z}}{\underline{z}} < \varepsilon_2$, update $\underline{z}'$ with the (MP) objective, otherwise, update $\underline{z}$ with the (MP) objective.

**Step 2.** Solve $\left(\overline{\text{SP}}\right)$ in each scenario with first-stage solutions $\boldsymbol{w}$ and $\boldsymbol{w}'$:

(a) Define $\left(\overline{\text{SP}}\right)$ in all scenarios $s \in \mathscr{S}$ and perturb its constraints (Equation (B.10)).

(b) Solve $\left(\overline{\text{SP}}\right)$ to derive $\boldsymbol{\Pi}_s$ for each $s \in \mathscr{S}$; If $\left(\overline{\text{SP}}\right)$ is infeasible, add Benders' feasibility cut (Equation (B.8)) to $\Theta^F$, otherwise, add dual integer cut (Equation (6)) to $\Theta^D$.

(c) Update $\tilde{z} \leftarrow \min\left\{\tilde{z},\ \rho\sum_{i\in\mathscr{F}}\sum_{t\in\tilde{\mathscr{T}}_i^{\text{dep}}} g_{it}(w_{i,t-1}^{\text{dep}} - w_{it}^{\text{dep}}) + (1-\rho)\sum_{s\in\mathscr{S}} p_s \bar{\Psi}(\boldsymbol{w}, \boldsymbol{\phi}_s)\right\}$ (or similarly with (MP) solution $\boldsymbol{w}'$); If $\frac{\tilde{z}-\underline{z}'}{\underline{z}'} \geq \varepsilon_1$ and $r \leq r^*$, go to Step 1.

**Termination.** Generate final (IMSOAN) solution:

(a) Solve (MP) with all cuts from $\Theta^D$ and $\Theta^F$—without neighborhood constraints; Update global lower bound $\underline{z}$ as the (MP) objective; Output (MP) solution $\boldsymbol{w}^*$.

(b) Solve (SP) in all scenarios $s \in \mathscr{S}$ with $\boldsymbol{w}^*$; If (SP) is infeasible in scenario $s$, add integer feasibility cut (7) to $\Theta^F$, go to Step 3(a); otherwise update global upper bound $\overline{z} \leftarrow \rho\sum_{i\in\mathscr{F}}\sum_{t\in\tilde{\mathscr{T}}_i^{\text{dep}}} g_{it}(w_{i,t-1}^{\text{dep}*} - w_{it}^{\text{dep}*}) + (1-\rho)\sum_{s\in\mathscr{S}} p_s \Psi(\boldsymbol{w}^*, \boldsymbol{\phi}_s)$; output (SP) solutions $(\boldsymbol{x}_s^*)_{s\in\mathscr{S}}$.

cut generation from global exploration to local exploitation. Ultimately, they enhance the computational efficiency of the master problem and accelerate convergence of the algorithm.

## 5. Scenario Generation

The computational and practical performance of any stochastic program depends critically on the accuracy and size of its scenario set. In terms of accuracy, misestimates of scenario probabilities can result in sub-optimal first-stage decisions. In terms of size, small scenario sets may not capture the full range of second-stage outcomes that are relevant to first-stage decisions, whereas large scenario sets may lead to computational intractability. To address this latter issue, Römisch (2009) proposed a scenario reduction approach that starts from a "true" probability distribution and, using

heuristic algorithms, generates an approximate distribution of prescribed cardinality—dictated by the computational requirements of the stochastic program. However, this approach is not applicable to our (IMSOAN), for two reasons. First, the exhaustive scenario set is of very large cardinality, with $2^{|\mathscr{K}||\mathscr{T}|}$ possible observations of the parameters $\Phi_{kt}$. Second, and most importantly, estimates of the probability of each of these $2^{|\mathscr{K}||\mathscr{T}|}$ scenarios are unavailable—thus raising the issue of the accuracy of the scenario set. To circumvent these challenges, we propose a data-driven scenario generation approach to construct a scenario set of prescribed cardinality, and their probabilities, from historical records of operations—using exact integer programming.

We start with a set $\mathscr{S}'$ of scenario realizations, obtained from historical data. Let $S' = |\mathscr{S}'|$. Let $\boldsymbol{p}' = \{p'_1, \cdots, p'_{S'}\}$ be a probability vector over $\mathscr{S}'$. In our setting, each scenario realization corresponds to a day of operating conditions, that is, to values of $\Phi_{kt}$ across all airports for all time periods. We assume that all observations have equal probabilities, so $p'_{s'} = \frac{1}{|\mathscr{S}'|}$ for all $s' \in \mathscr{S}'$. Our scenario generation procedure selects a subset $\mathscr{S} \subset \mathscr{S}'$ such that $|\mathscr{S}| = S \leq S'$, and maps each of the $S'$ original scenarios into one of the $S$ selected scenarios. This mapping is then used to define a probability vector over $\mathscr{S}$, denoted by $\boldsymbol{p} = \{p_1, \cdots, p_S\}$.

The objective is to minimize the *Kantorovich distance* between the probability vectors $\boldsymbol{p}'$ and $\boldsymbol{p}$. Let us first consider a symmetric distance using the $L^1$-norm: $\Delta_1(s_1, s_2) = \sum_{k\in\mathscr{K}} \sum_{t\in\mathscr{T}} |\phi_{kts_1} - \phi_{kts_2}|, \forall s_1, s_2 \in \mathscr{S}'$. The Kantorovich distance, denoted by $D(\boldsymbol{p}', \boldsymbol{p})$, is defined as the weighted distance, over all scenarios in $\mathscr{S}'$, to their closest one within the selected set $\mathscr{S}$:

$$D(\boldsymbol{p}', \boldsymbol{p}) = \sum_{s'\in\mathscr{S}'} p'_{s'} \min_{s\in\mathscr{S}} \Delta_1(s', s). \tag{9}$$

We cast the scenario generation problem as a *p-median problem* (Reese 2006). Each scenario $s' \in \mathscr{S}'$ is viewed as a "demand point" and each selected scenario $s \in \mathscr{S}$ is viewed as a "facility". The "distance" between $s$ and $s'$ is defined as $d_{s',s} = p'_{s'} \Delta_1(s', s)$. The decision variables include: (i) a binary variable $y_s$ equal to 1 if scenario $s \in \mathscr{S}'$ is included in the subset $\mathscr{S}$ (i.e.,"facility $s$ is built"), and 0 otherwise, and (ii) a binary variable $x_{s',s}$ equal to 1 if the probability of $s' \in \mathscr{S}'$ is assigned to that of $s \in \mathscr{S}$ (i.e.,"demand point $s'$ is served by facility $s$"), and 0 otherwise. The scenario generation problem is then formulated as:

$$\text{(SG)} \qquad \min \sum_{s'\in\mathscr{S}'} \sum_{s\in\mathscr{S}'} d_{s',s} x_{s',s}, \tag{10a}$$

$$\text{s.t.} \sum_{s\in\mathscr{S}'} y_s \leq S, \tag{10b}$$

$$x_{s',s} \leq y_s, \quad \forall s', s \in \mathscr{S}', \tag{10c}$$

$$\sum_{s\in\mathscr{S}'} x_{s',s} = 1, \quad \forall s' \in \mathscr{S}', \tag{10d}$$

$$y_s \in \{0,1\} \quad \forall s \in \mathscr{S}', \tag{10e}$$

$$x_{s',s} \in \{0,1\} \quad \forall s', s \in \mathscr{S}'. \tag{10f}$$

Equation (10a) minimizes the Kantorovich distance. Constraint (10b) enforces that at most $S$ scenarios are selected in $\mathscr{S}$. Constraint (10c) states that all scenarios $s' \in \mathscr{S}'$ are mapped to those selected in $\mathscr{S}$. Constraint (10d) guarantees that each scenario $s'$ is assigned to exactly one scenario in $\mathscr{S}$. Constraints (10e) and (10f) define the domain of the variables.

We obtain the scenario set $\mathscr{S} = \{s \in \mathscr{S}' | y^*_s = 1\}$, and construct the distribution $\boldsymbol{p}$ as follows:

$$p_s = \sum_{s' \in \mathscr{S}'} p'_{s'} x^*_{s',s}.$$

Proposition 3 (proved in Appendix C) states that all scenarios selected in $\mathscr{S}$ are "mapped" into themselves, and that exactly $S$ scenarios are selected. From this result, we tighten the formulation of (SG) by replacing Equation (10b) with $\sum_{s \in \mathscr{S}'} y_s = S$ and Equation (10d) with $\sum_{s \in \mathscr{S}'} x_{s',s} = 1 - y_{s'}, \forall s' \in \mathscr{S}'$. Computationally, (SG) can effectively tackle with up to 2,000 scenarios in $\mathscr{S}'$.

PROPOSITION 3. *If $d_{s'_1,s'_2} > 0$ for all $s'_1 \neq s'_2 \in \mathscr{S}'$ then we have, at the optimum of (SG):*
*(i) $x^*_{s,s} = 1$ for each $s \in \mathscr{S}'$ such that $y^*_s = 1$, and*
*(ii) $\sum_{s \in \mathscr{S}'} y^*_s = S$ (i.e., Constraint (10b) is binding).*

Scenario generation is applied twice in our algorithm. First and foremost, it determines the scenario set $\mathscr{S}$ as input to our model. Second, it reduces the scenario set into an even smaller one to solve the LP relaxation of (IMSOAN) and derive the core point $\boldsymbol{w}^{(o)}$ (see Algorithm 1).

## 6. Computational Results

We implement the proposed model and algorithm using real-world data from the US National Airspace System. We use CPLEX 12.5 as our IP and LP solver with C♯ (Visual Studio 2015) on a workstation with a 32-core Intel Xeon CPU (3.0 GHz) and 128 GB RAM.

### 6.1. Input Data and Model Instances

We use scheduling data from 2007. In that year, no demand management measures were in place at even the busiest US airports, so the schedule of flights can be used as a reasonable proxy of airlines' preferred schedules. Given the low variability in flight schedules from one day to the next, we use scheduling inputs from one day, September 18, 2007. We consider networks of increasing size comprising some to all of the "CORE 30" airports (the 30 most prominent US airports). Specifically, we first consider test instances with the 6 airports subject to the highest average delays (i.e., ALT, EWR, JFK, LGA, ORD and PHL). We then increase the number of airports to 12, 18 and up to including all CORE 30 airports. Ultimately, this setup demonstrates the applicability of (IMSOAN) on a network of the size of the US National Airspace System.

Flight schedules, aircraft connection and weather data are obtained from the Aviation System Performance Metrics (ASPM) database (Federal Aviation Administration 2017). We use minimum

aircraft turnaround times from Pyrgiotis (2011) and passenger connections from Barnhart et al. (2014). We calibrate $\Delta_i^{\min}$ and $\Delta_i^{\max}$ by assuming that each flight's block-time can be lower or higher than the scheduled one by 10%. We use the VMC and IMC capacity envelopes at the three airports in the New York area (i.e., JFK, EWR and LGA) from Simaiakis (2012). At the other 27 airports, we approximate the VMC and IMC capacity envelopes with three segments refecting the arrival, departure and total capacities estimated by the Federal Aviation Administration (2004).

We set the maximum displacement $\delta$ of all flights as one 15-minute period. The unit costs of departure and arrival delay per period of flight $i \in \mathscr{F}$ are set to $c_i^{\text{dep}} = 1$ and $c_i^{\text{arr}} = 1.2$ respectively, reflecting that departure delays are less expensive and less environmentally damaging to operate than arrival delays. We vary the parameter $\rho$ used to weight schedule displacement vs. delay costs by defining four values: $\rho_1 = 0.46, \rho_2 = 0.67, \rho_3 = 0.82, \rho_4 = 0.95$.

We obtain the maximum departure and arrival delays $l_i^{\text{dep}}$ and $l_i^{\text{arr}}$ allowed for each flight $i \in \mathscr{F}$ from the MAGHO solution in the worst-case operating scenario. Specifically, let $v_i^{\text{dep}}$ and $v_i^{\text{arr}}$ be the departure and arrival delays of each flight $i \in \mathscr{F}$ from the optimal MAGHO solution without the maximum delay restriction, when all airports operate under IMC in each period (Equations (2a)–(2e) and (2h)–(2o)). Then, we set $l_i^{\text{dep}} = v_i^{\text{dep}} + 1$ and $l_i^{\text{arr}} = v_i^{\text{arr}} + 1$ for all $i \in \mathscr{F}$ into (IMSOAN).

To obtain the scenario set $\mathscr{S}$ and the probability distribution $(p_s)_{s \in \mathscr{S}}$, we apply our scenario generation approach (Section 5) with historical weather data over a 5-year period. Each of the 1,826 days defines a scenario $s' \in \mathscr{S}'$ occurring with probability $p'_{s'} = \frac{1}{|\mathscr{S}'|}$.

**Table 2 Test instances and model size.**

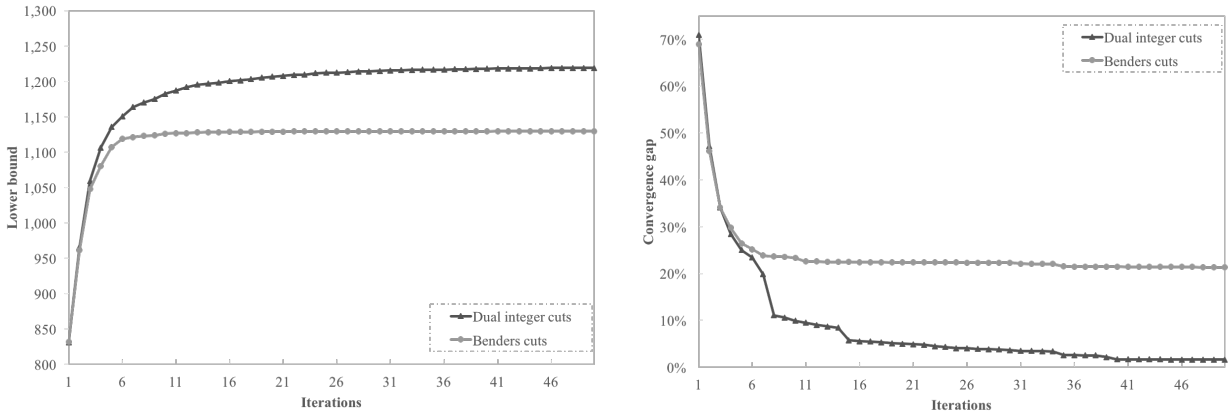
| Instance | | | | | | Size of (IMSOAN) | | | |
|---|---|---|---|---|---|---|---|---|---|
| ID | $F$ | $\lvert\mathscr{C}^1\rvert$ | $\lvert\mathscr{C}^2\rvert$ | $\sum_i(\lvert\mathscr{T}_i^{\text{dep}}\rvert + \lvert\mathscr{T}_i^{\text{arr}}\rvert)$ | $S$ | Binary var. | Integer var. | Total var. | Total con. |
| I_K6_S3 | 14,611 | 68,066 | 9,195 | 140,377 | 3 | 479,575 | 43,833 | 523,408 | 1,278,664 |
| I_K6_S5 | | | | | 5 | 760,329 | 73,055 | 833,384 | 2,037,026 |
| I_K6_S10 | | | | | 10 | 1,462,214 | 146,110 | 1,608,324 | 3,932,931 |
| I_K6_S30 | | | | | 30 | 4,269,754 | 438,330 | 4,708,084 | 11,516,551 |
| I_K12_S3 | 19,949 | 104,648 | 13,199 | 177,360 | 3 | 611,876 | 59,847 | 671,723 | 1,676,628 |
| I_K12_S5 | | | | | 5 | 966,596 | 99,745 | 1,066,341 | 2,658,118 |
| I_K12_S10 | | | | | 10 | 1,853,396 | 199,490 | 2,052,886 | 5,111,843 |
| I_K12_S30 | | | | | 30 | 5,400,596 | 598,470 | 5,999,066 | 14,926,743 |
| I_K18_S3 | 24,385 | 136,105 | 16,983 | 207,232 | 3 | 719,236 | 73,155 | 792,391 | 2,005,395 |
| I_K18_S5 | | | | | 5 | 1,133,700 | 121,925 | 1,255,625 | 3,170,305 |
| I_K18_S10 | | | | | 10 | 2,169,860 | 243,850 | 2,413,710 | 6,082,580 |
| I_K18_S30 | | | | | 30 | 6,314,500 | 731,550 | 7,046,050 | 17,731,680 |
| I_K30_S3 | 29,058 | 166,062 | 20,840 | 239,618 | 3 | 835,086 | 87,174 | 922,260 | 2,358,114 |
| I_K30_S5 | | | | | 5 | 1,314,322 | 145,290 | 1,459,612 | 3,722,622 |
| I_K30_S10 | | | | | 10 | 2,512,412 | 290,580 | 2,802,992 | 7,133,892 |
| I_K30_S30 | | | | | 30 | 7,304,772 | 871,740 | 8,176,512 | 20,778,972 |

We divide the day between 6 a.m. and 12 a.m. into 72 15-minute periods, plus 15 extra periods to ensure feasibility. We generate 16 test instances, each with an ID "I_Ka_Sb" where "a" and "b" denote the number of airports in $\mathscr{K}$ and the number of scenarios in $\mathscr{S}$, respectively. Table 2 reports the number of inputs, variables and constraints in each instance. Note the sheer size of (IMSOAN). The largest instance, I_K30_S30, results in 8.2 million variables and 20.8 million constraints. When more than 30 scenarios are included, we run into memory limitations; nonetheless, our solution algorithm can tackle such instances on higher-memory machines.

### 6.2. Convergence of Dual Integer Cut

We investigate the convergence of our stochastic integer programming procedure with the dual integer cut (Section 4.2). For unbiased assessments, we do not adopt any acceleration strategy here.

Figure 3 shows the lower bound $\underline{z}$ (Figure 3a) and the convergence gap $\frac{\tilde{z}-\underline{z}}{\underline{z}}$ (Figure 3b), with our dual integer cuts and the Benders cuts, in the largest test instance. The plots show that the dual integer cuts significantly outperform the Benders cuts by improving the lower bound $\underline{z}$ as well as the "relaxation upper bound" $\tilde{z}$ (suggested by the larger relative difference in the convergence gap than in the lower bound). With the dual integer cuts, the convergence gap gets close to 1% after 35 iterations; in contrast, it does not even reach 20% after 50 iterations with the Benders cuts. This suggests that the additional terms in Equation (6) are instrumental to tighten the feasible region of Problem (MP) and improve the algorithm's performance. Note, finally, the long-tail convergence of the algorithm, which motivates our neighborhood constraints.




(a) Lower bound (b) Convergence gap

**Figure 3 Lower bound and convergence gap over 50 iterations for Instance I_K30_S30 with $\rho = \rho_2$.**

Table 3 shows similar convergence results in different test instances with 6 airports (top) and 30 airports (bottom) and 5, 10 and 30 scenarios. The table underscores the robustness of our result: in all cases, the dual integer cuts result in significantly lower convergence gaps than the

**Table 3 Convergence gap in different instances of 6 airports and 30 airports with $\rho = \rho_2$.**

| Iter. | I_K6_S5 | | I_K6_S10 | | I_K6_S30 | |
|---|---|---|---|---|---|---|
| | Dual integer cuts | Benders cuts | Dual integer cuts | Benders cuts | Dual integer cuts | Benders cuts |
| 10 | 7.80% | 11.57% | 7.50% | 13.56% | 5.84% | 11.18% |
| 20 | 4.63% | 11.16% | 4.26% | 13.04% | 2.88% | 10.82% |
| 30 | 2.47% | 11.11% | 3.07% | 12.98% | 1.94% | 10.80% |
| 40 | 2.14% | 11.09% | 2.43% | 12.95% | 1.61% | 10.79% |
| 50 | 1.87% | 11.09% | 2.36% | 12.95% | 1.50% | 10.79% |
| 60 | 1.78% | 11.09% | 1.99% | 12.95% | 1.36% | 10.79% |
| 70 | 1.71% | 11.09% | 1.90% | 12.95% | 1.29% | 10.79% |
| Iter. | I_K30_S5 | | I_K30_S10 | | I_K30_S30 | |
| | Dual integer cuts | Benders cuts | Dual integer cuts | Benders cuts | Dual integer cuts | Benders cuts |
| 10 | 19.29% | 37.00% | 9.98% | 22.68% | 9.90% | 23.31% |
| 20 | 12.52% | 36.04% | 5.71% | 21.61% | 4.99% | 22.37% |
| 30 | 6.99% | 35.79% | 4.18% | 21.36% | 3.60% | 22.32% |
| 40 | 4.76% | 35.61% | 2.66% | 21.35% | 1.67% | 21.44% |
| 50 | 4.15% | 34.53% | 1.97% | 20.74% | 1.58% | 21.29% |
| 60 | 3.44% | 34.49% | 1.85% | 20.71% | 1.54% | 21.26% |
| 70 | 2.75% | 34.45% | 1.48% | 20.70% | 1.47% | 21.26% |

Benders cuts. Interestingly, the benefits of the dual integer cuts increase with the model size. Indeed, convergence is similar with 6 and 30 airports with dual integer cuts (although each iteration obviously involves longer runtimes with 30 airports); in comparison, the performance of the Benders cut sharply deteriorates as the model gets larger. Similarly, the convergence of the dual integer cut does not degenerate with more scenarios. These results underscore the performance of the proposed algorithm, as well as its potential to deal with even larger instances with high-memory machines.

### 6.3. Computational Performance of Acceleration Strategies

To test the acceleration strategies, we start from a baseline with the dual integer cuts only, referred to as "DIC". We add local branching and Pareto-optimality cuts ("DIC+LOB+POC"). Last, we add the neighborhood constraints—which corresponds to Algorithm 1 ("DIC+LOB+POC+NS"). Table 4 compares the three methods with 6 and 30 airports, 5, 10 and 30 scenarios, and different values of $\rho$. The table reports, upon termination, the lower bound ($\underline{z}$), the convergence gap $\left(\frac{\bar{z}-\underline{z}}{\underline{z}}\right)$, the number of iterations ($r$), and the total CPU time in minutes ($T^{\text{tot}}$).

First, comparisons between "DIC" and "DIC+LOB+POC" show that local branching and Pareto-optimality cuts reduce the convergence gap from 2.67% to 2.07% on average. These improvements are stronger when both methods terminates in the same number of iterations (e.g., Instance I_K6_S5 with $\rho = \rho_1$). However, these acceleration strategies do not significantly reduce runtimes, because (MP) becomes more time-consuming when more cuts are added at each iteration.

**Table 4 Effectiveness of the acceleration strategies.**

| Instance | | DIC | | | | DIC+LOB+POC | | | | DIC+LOB+POC+NC | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | $\rho$ | $\underline{z}$ | $\frac{\bar{z}-\underline{z}}{\underline{z}}$ | $r$ | $T^{\text{tot}}$ | $\underline{z}$ | $\frac{\bar{z}-\underline{z}}{\underline{z}}$ | $r$ | $T^{\text{tot}}$ | $\underline{z}$ | $\frac{\bar{z}-\underline{z}}{\underline{z}}$ | $r$ | $T^{\text{tot}}$ |
| I_K6_S5 | $\rho_1$ | 2,266 | 4.71% | 70 | 664 | 2,293 | 2.56% | 70 | 1,213 | 2,306 | 1.53% | 70 | 692 |
| | $\rho_2$ | 1,542 | 1.72% | 70 | 703 | 1,554 | 0.95% | 30 | 224 | 1,555 | 0.99% | 39 | 189 |
| | $\rho_3$ | 883 | 0.98% | 28 | 99 | 886 | 0.88% | 14 | 45 | 884 | 0.95% | 13 | 24 |
| | $\rho_4$ | 286 | 0.43% | 3 | 2 | 286 | 0.32% | 3 | 4 | 286 | 0.42% | 3 | 2 |
| I_K6_S10 | $\rho_1$ | 1,950 | 4.87% | 70 | 1,828 | 1,959 | 3.82% | 70 | 3,257 | 1,980 | 1.66% | 70 | 1,555 |
| | $\rho_2$ | 1,345 | 1.92% | 70 | 1,592 | 1,354 | 1.00% | 70 | 2,023 | 1,354 | 1.00% | 70 | 734 |
| | $\rho_3$ | 769 | 0.94% | 29 | 177 | 769 | 1.00% | 37 | 328 | 776 | 0.97% | 28 | 98 |
| | $\rho_4$ | 250 | 0.51% | 3 | 10 | 250 | 0.47% | 3 | 9 | 250 | 0.57% | 3 | 10 |
| I_K6_S30 | $\rho_1$ | 2,383 | 2.96% | 70 | 4,397 | 2,390 | 2.08% | 70 | 4,427 | 2,391 | 1.14% | 70 | 2,757 |
| | $\rho_2$ | 1,612 | 1.29% | 70 | 4,497 | 1,616 | 1.00% | 51 | 4,013 | 1,617 | 0.98% | 54 | 2,763 |
| | $\rho_3$ | 921 | 0.96% | 20 | 398 | 921 | 1.00% | 20 | 388 | 921 | 0.96% | 20 | 241 |
| | $\rho_4$ | 299 | 0.21% | 3 | 19 | 298 | 0.41% | 3 | 33 | 298 | 0.41% | 3 | 29 |
| I_K30_S5 | $\rho_1$ | 1,035 | 16.42% | 70 | 1,356 | 1,042 | 12.17% | 70 | 1,447 | 1,042 | 12.17% | 70 | 1,447 |
| | $\rho_2$ | 801 | 2.75% | 70 | 1,456 | 806 | 1.86% | 70 | 1,254 | 807 | 1.26% | 70 | 894 |
| | $\rho_3$ | 484 | 1.00% | 33 | 418 | 484 | 0.95% | 27 | 306 | 484 | 0.78% | 26 | 251 |
| | $\rho_4$ | 164 | 0.41% | 3 | 14 | 164 | 0.40% | 3 | 14 | 164 | 0.40% | 3 | 11 |
| I_K30_S10 | $\rho_1$ | 1,778 | 8.67% | 70 | 2,548 | 1,767 | 7.12% | 70 | 2,627 | 1,767 | 7.12% | 70 | 2,627 |
| | $\rho_2$ | 1,274 | 1.48% | 70 | 2,623 | 1,277 | 1.37% | 70 | 2,361 | 1,278 | 1.20% | 70 | 1,524 |
| | $\rho_3$ | 739 | 0.88% | 20 | 256 | 739 | 0.92% | 17 | 247 | 740 | 0.98% | 22 | 107 |
| | $\rho_4$ | 243 | 0.51% | 3 | 24 | 243 | 0.36% | 3 | 23 | 243 | 0.36% | 3 | 22 |
| I_K30_S30 | $\rho_1$ | 1,696 | 7.37% | 70 | 7,832 | 1,679 | 6.21% | 70 | 7,050 | 1,679 | 6.21% | 70 | 7,050 |
| | $\rho_2$ | 1,220 | 1.47% | 70 | 7,662 | 1,221 | 1.37% | 70 | 6,658 | 1,222 | 1.23% | 70 | 4,730 |
| | $\rho_3$ | 706 | 1.00% | 15 | 646 | 705 | 0.99% | 14 | 594 | 706 | 0.98% | 18 | 342 |
| | $\rho_4$ | 232 | 0.57% | 3 | 68 | 232 | 0.51% | 3 | 67 | 232 | 0.51% | 3 | 54 |
| *Average:* | | *1,037* | *2.67%* | *42* | *1,637* | *1,039* | *2.07%* | *39* | *1,609* | *1,041* | *1.87%* | *39* | *1,173* |

Then, our neighborhood constraints further improve algorithmic performance, as compared to traditional acceleration techniques. They reduce average CPU times from 1,609 to 1,173 minutes, by restricting the solution space of (MP) in later iterations. And they reduce the average convergence gap upon termination from 2.07% to 1.87%, by making the dual integer cuts more tailored to the "relevant" first-stage solutions (that is, to the flights that might be displaced). This neighborhood-based approach can thus accelerate decomposition algorithms for two-stage stochastic programs that optimize a small subset of variables over a large set of variables.

Finally, convergence is faster for larger values of $\rho$. Indeed, for large values of $\rho$, the first-stage schedule displacement is smaller and fewer cuts need to be generated to evaluate the recourse function. Still, our algorithm can handle small values of $\rho$, albeit at the cost of longer CPU times. We further discuss the effect of the weight $\rho$ on solution quality in Section 6.5.

### 6.4. Solution Quality: Comparison with CPLEX

We now compare the solution from our algorithm to the one obtained by solving (IMSOAN) directly with CPLEX. Table 5 reports, with $\rho = \rho_2$ and $\rho = \rho_4$, the optimal solution ($z^*$), the CPLEX

**Table 5 Comparison of the algorithm with direct CPLEX implementation.**

| Instance | | CPLEX | | Algorithm 1 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\rho$ | ID | $z^*$ | $T^{\text{cpx}}$ | $\overline{z}$ | $\underline{z}$ | $T^{\text{sol}}$ | $T^{\text{tot}}$ | $\frac{\overline{z}-\underline{z}}{\underline{z}}$ | $\frac{\overline{z}-z^*}{z^*}$ | $\frac{\overline{z}-\underline{z}}{\overline{z}}$ |
| | I_K6_S3 | 752 | 61 | 758 | 714 | 54 | 157 | 1.08% | 0.90% | 5.82% |
| $\rho_2$ | I_K6_S5 | 1,593 | 159 | 1,605 | 1,555 | 74 | 189 | 0.99% | 0.74% | 3.10% |
| | I_K6_S30 | —— | —— | 1,673 | 1,617 | 1,466 | 2,763 | 0.98% | —— | 3.35% |
| | I_K12_S3 | 741 | 57 | 753 | 709 | 82 | 539 | 1.77% | 1.58% | 5.85% |
| $\rho_2$ | I_K12_S5 | 664 | 181 | 671 | 635 | 125 | 620 | 1.21% | 1.03% | 5.36% |
| | I_K12_S30 | —— | —— | 1,491 | 1,429 | 1,521 | 2,969 | 1.58% | —— | 4.13% |
| | I_K18_S3 | 785 | 53 | 796 | 754 | 111 | 547 | 1.48% | 1.38% | 5.27% |
| $\rho_2$ | I_K18_S5 | 829 | 587 | 837 | 793 | 210 | 604 | 1.24% | 0.95% | 5.30% |
| | I_K18_S30 | —— | —— | 1,415 | 1,363 | 1,708 | 3,889 | 1.02% | —— | 3.67% |
| | I_K30_S3 | 722 | 57 | 734 | 692 | 127 | 550 | 1.93% | 1.61% | 5.67% |
| $\rho_2$ | I_K30_S5 | 842 | 601 | 851 | 807 | 250 | 894 | 1.26% | 1.12% | 5.22% |
| | I_K30_S30 | —— | —— | 1,279 | 1,222 | 2,061 | 4,730 | 1.23% | —— | 4.46% |
| | *Average:* | —— | —— | *1,072* | *1,024* | *649* | *1,538* | *1.31%* | *1.16%* | *4.77%* |
| | I_K6_S3 | 149 | 41 | 149 | 141 | 1 | 1 | 0.00% | 0.00% | 5.43% |
| $\rho_4$ | I_K6_S5 | 293 | 115 | 294 | 286 | 1 | 2 | 0.42% | 0.37% | 2.84% |
| | I_K6_S30 | —— | —— | 307 | 298 | 25 | 29 | 0.41% | —— | 2.80% |
| | I_K12_S3 | 147 | 47 | 147 | 140 | 1 | 1 | 0.00% | 0.00% | 5.01% |
| $\rho_4$ | I_K12_S5 | 136 | 151 | 136 | 128 | 1 | 3 | 0.00% | 0.00% | 5.99% |
| | I_K12_S30 | —— | —— | 274 | 266 | 28 | 33 | 0.45% | —— | 3.16% |
| | I_K18_S3 | 156 | 53 | 156 | 149 | 2 | 3 | 0.00% | 0.00% | 4.78% |
| $\rho_4$ | I_K18_S5 | 163 | 183 | 163 | 155 | 4 | 5 | 0.09% | 0.00% | 4.88% |
| | I_K18_S30 | —— | —— | 261 | 253 | 31 | 39 | 0.49% | —— | 3.23% |
| | I_K30_S3 | 146 | 56 | 146 | 138 | 2 | 3 | 0.00% | 0.00% | 5.48% |
| $\rho_4$ | I_K30_S5 | 172 | 376 | 173 | 164 | 9 | 11 | 0.40% | 0.44% | 5.48% |
| | I_K30_S30 | —— | —— | 242 | 232 | 48 | 54 | 0.51% | —— | 3.80% |
| | *Average:* | —— | —— | *204* | *196* | *13* | *15* | *0.23%* | *0.10%* | *4.41%* |

"——" indicates that no feasible solution is obtained from direct CPLEX implementation.

solution time in minutes ($T^{\text{cpx}}$), and our algorithm's outputs: the solution $\overline{z}$, the lower bound $\underline{z}$, the solution time in minutes $T^{\text{sol}}$, the CPU time in minutes $T^{\text{tot}}$,[4] the convergence gap $\frac{\tilde{z}-\underline{z}}{\underline{z}}$, the optimality gap $\frac{\overline{z}-z^*}{z^*}$ and the lower bound gap $\frac{\overline{z}-\underline{z}}{\overline{z}}$. The optimality gap compares our solution to the optimum; the lower bound gap provides solution quality guarantees when the true optimum cannot be obtained with CPLEX.

CPLEX does not even provide a feasible solution in instances with more than 5 scenarios. In contrast, our algorithm consistently yields high-quality solutions in instances with 30 scenarios.

Among the small-scale instances that can be solved by CPLEX, our algorithm yields near-optimal solutions in shorter runtimes. When $\rho = \rho_4$, it returns optimal solutions in most cases, and is 34 times faster than CPLEX in the largest instance (Instance I_K30_S5). When $\rho = \rho_2$, the algorithm

[4] The CPU time is equal to the solution time minus processing and presolving times. To save memory, we clean the formulations of (MP) and (SP) at each iteration. In practice, processing and presolving times can be largely reduced.

returns close-top-optimal solutions, with an average optimality gap is 1.16%; it increases the CPU time as compared to CPLEX but still reduces the solution time by 30-60%.

Among the large-scale instances, we use two measures to assess solution quality. First, the lower bound gap $\frac{\overline{z}-\underline{z}}{\overline{z}}$ ranges from 2.80% to 4.46%. The true optimality gap falls below these values, thus guaranteeing near-optimality of the solutions. Second, note that, in small-scale instances, the convergence gap $\frac{\tilde{z}-\underline{z}}{\underline{z}}$ is consistently close to the true optimality gap—so the convergence gap can be seen as an approximation of the true optimality gap. In large-scale instances, the convergence gap lies consistently within 0.4-0.5% for $\rho=\rho_4$ and within 0.9-1.6% for $\rho=\rho_2$, thus suggesting that the algorithm's solutions are close to the true (unknown) optimum.

### 6.5. Benefits of Stochastic Optimization

To investigate the benefits of stochastic optimization, Table 6 reports the value of the stochastic solution (VSS) and the expected value of perfect information (EVPI) (Birge and Louveaux 2011). Specifically, the table lists: (i) the solution of the recourse problem (RP) obtained from Algorithm 1, (ii) the expected value solution (EEV) resulting from a deterministic problem with a single nominal scenario, and (iii) the wait-and-see solution (WS) resulting from a deterministic problem based on perfect information. The VSS and the EVPI are then obtained as:

$$\text{VSS}=\text{EEV}-\text{RP}, \quad \text{and} \quad \text{EVPI}=\text{RP}-\text{WS}.$$

VSS quantifies the benefits of the stochastic optimization framework developed in this paper, as compared to a simpler approach that would directly optimize first-stage and second-stage solutions with a single "representative" scenario (i.e., by considering the average capacity at each airport in each time period). EVPI quantifies the residual cost of uncertainty resulting from the fact that the first-stage scheduling decisions need to be made before uncertainty is resolved (as opposed to optimizing the first-stage and second-stage solutions in each scenario independently).

First and foremost, the value of the stochastic solution is significant in all test instances—with the ratio $\frac{\text{VSS}}{\text{RP}}$ ranging from 4.0% to 23.1%. This underscores that our stochastic optimization approach can significantly improve the first-stage scheduling interventions solution, as compared to a baseline approach that ignores operating uncertainty. Meanwhile, the expected value of perfect information (EVPI) is also significant—with the ratio $\frac{\text{EVPI}}{\text{RP}}$ ranging from 3.8% to 26.8%. This is not surprising, as the optimal balance of schedule displacement and flight delays is highly sensitive to the stochastic airport operating conditions. Nonetheless, our two-stage stochastic programming approach bridges a significant portion of the gap between the baseline solution that ignores capacity uncertainty (EEV) and the solution obtained under perfect information (WS)—reflected by the ratio $\frac{\text{VSS}}{\text{VSS}+\text{EVPI}}$ ranging between 42.2% and 70.2%.

**Table 6 The value of stochastic solution and the expected value of perfect information.**

| Instance | | | Value of Stochastic Solution (VSS) | | | Expected Value of Perfect Information (EVPI) | | | |
|---|---|---|---|---|---|---|---|---|---|
| ID | $\rho$ | RP | EEV | VSS | $\frac{\text{VSS}}{\text{RP}}$ | WS | EVPI | $\frac{\text{EVPI}}{\text{RP}}$ | $\frac{\text{VSS}}{\text{VSS+EVPI}}$ |
| I_K30_S5 | $\rho_1$ | 1,214 | 1,495 | 281 | 23.13% | 889 | 325 | 26.76% | 46.37% |
| | $\rho_2$ | 851 | 1,038 | 187 | 21.91% | 772 | 79 | 9.29% | 70.21% |
| | $\rho_3$ | 510 | 561 | 51 | 9.93% | 475 | 36 | 6.97% | 58.75% |
| | $\rho_4$ | 173 | 181 | 8 | 4.86% | 163 | 10 | 6.06% | 44.50% |
| I_K30_S10 | $\rho_1$ | 1,937 | 2,298 | 361 | 18.65% | 1,531 | 406 | 20.96% | 47.08% |
| | $\rho_2$ | 1,328 | 1,532 | 205 | 15.42% | 1,213 | 115 | 8.68% | 64.00% |
| | $\rho_3$ | 770 | 834 | 64 | 8.26% | 713 | 57 | 7.42% | 52.68% |
| | $\rho_4$ | 251 | 261 | 10 | 4.04% | 237 | 14 | 5.55% | 42.16% |
| I_K30_S30 | $\rho_1$ | 1,837 | 2,197 | 361 | 19.63% | 1,419 | 418 | 22.76% | 46.30% |
| | $\rho_2$ | 1,279 | 1,464 | 185 | 14.45% | 1,146 | 133 | 10.37% | 58.21% |
| | $\rho_3$ | 738 | 796 | 58 | 7.87% | 691 | 47 | 6.36% | 55.30% |
| | $\rho_4$ | 242 | 253 | 11 | 4.75% | 232 | 9 | 3.77% | 55.76% |

Finally, both $\frac{\text{VSS}}{\text{RP}}$ and $\frac{\text{EVPI}}{\text{RP}}$ become larger as $\rho$ becomes smaller: when a higher weight is assigned to the second-stage delay cost, information uncertainty in the second stage has a more significant impact on the first-stage decisions. Combined with Section 6.4, these results underscore that larger values of $\rho$ (which result in a smaller schedule displacement) induce faster algorithmic convergence but more moderate benefits of our stochastic optimization approach—while the reverse holds for smaller values of $\rho$ (which result in a larger schedule displacement).

### 6.6. Benefits of Scenario Generation

Recall that our algorithm can solve 30-scenario instances, while CPLEX can only handle up to 5 scenarios. We conclude by answering two questions. First, what are the benefits (if any) of near-optimal solutions with 30 scenarios vs. optimal solutions with 5 scenarios? Second, what are the benefits of our scenario generation approach vs. baseline scenario generation heuristics?

To answer these questions, we solve 9 variants of the model (with 30 airports and $\rho = \rho_2$), by considering 3 solution approaches and, for each one, 3 procedures to generate scenarios. The 3 solution approaches are: (i) CPEX with 5 scenarios, (ii) Algorithm 1 with 5 scenarios, and (iii) Algorithm 1 with 30 scenarios. The 3 scenario generation procedures are referred to as "H" ("Heuristic"), "H/SG" ("Heuristic/Scenario Generation") and "SG" ("Scenario Generation"). Procedure H is defined as follows. We rank all scenarios in the 5-year set $\mathscr{S}'$ by the total number of VMC periods across the entire day and the 30 airports (from the "worst-weather" scenario to the "best-weather" scenario). We then select "representative scenarios" by dividing the set of scenarios into 5 or 30 intervals, selecting the midpoint of each interval, and assigning equal probability to each of the selected scenarios. Procedure H/SG considers the same 5 or 30 representative scenarios but determines their probabilities by solving our model (SG)—where variables $(y_s)_{s\in\mathscr{S}'}$ are fixed. In contrast, Procedure SG selects the scenarios and their probabilities with (SG).

Table 7 compares the performance of the 9 resulting solutions to a baseline with the original schedule (i.e., no first-stage SI and optimal second-stage MAGHO). It reports the schedule displacement and the expected delay cost over the full set of 1,826 scenarios in the set $\mathscr{S}'$—that is, we perform out-of-sample evaluation by replicating MAGHO dynamics over all historical scenarios. The table also reports the total weighted cost and its percent-wise reduction from the baseline.

**Table 7 Performance of CPLEX and Algorithm 1, with Procedures H, H/SG and SG for scenario generation.**

| | | CPLEX, 5 scenarios | | | Algorithm 1, 5 scenarios | | | Algorithm 1, 30 scenarios | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Cost component | Baseline | H | H/SG | SG | H | H/SG | SG | H | H/SG | SG |
| Schedule displacement | 0 | 367 | 294 | 278 | 325 | 215 | 237 | 235 | 179 | 280 |
| Expected MAGHO cost | 5,177 | 3,781 | 3,789 | 3,797 | 3,900 | 3,965 | 3,900 | 3,849 | 3,886 | 3,610 |
| Total weighted cost | 1,726 | 1,505 | 1,459 | 1,451 | 1,517 | 1,465 | 1,458 | 1,440 | 1,415 | 1,390 |
| Total cost reduction | 0.00% | 12.78% | 15.44% | 15.92% | 12.11% | 15.10% | 15.51% | 16.56% | 18.01% | 19.45% |

The takeaways are twofold. First, applying Algorithm 1 with 30 scenarios results in significantly better solutions than applying CPLEX directly with only 5 scenarios—with corresponding total cost reductions of the order of 3-5%, regardless of the scenario generation procedure. This underscores the benefits of our algorithm, which enables to solve large-scale instances otherwise intractable. Second, our scenario generation approach enhances the performance of the model: as compared to Procedure H, Procedures H/SG and SG reduce total cost by 1.7-3.4% and by 3.4-3.8%, respectively. This underscores the benefits of selecting good scenarios and calibrating their probabilities to capture variations in operating conditions observed in historical data.

## 7. Case Study Results

Finally, (IMSOAN) provides insights on scheduling and operations across airport networks. Section 7.1 underscores the benefits of our integrated approach. Sections 7.2 and 7.3 identify where (i.e., at which airports) and when (i.e., at which times of day) scheduling interventions (SI) yield maximal efficiency. Unless otherwise specified, we consider Instance I_K30_S30.

### 7.1. Benefits of Integration

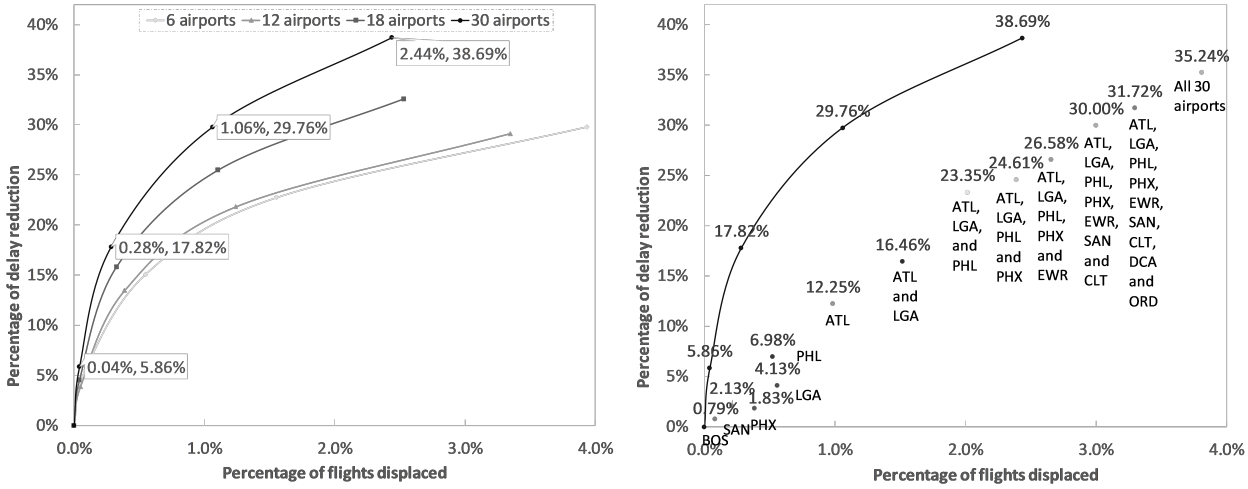
Figure 4 shows the trade-off between first-stage displacement and second-stage expected delays, obtained by varying $\rho$. Figure 4a shows results with 6, 12, 18 and 30 airports. Figure 4b compares our results with a sequential approach that first applies SI with schedule limits based on the VMC capacity of a subset $\mathscr{K}'$ of airports (Equations (11a)–(11c), where $Q_{kq}^{\text{VMC}}$ denotes the value of $Q_{kq}(\phi)$ in VMC), and then performs MAGHO with the resulting schedule (Equations (2a)–(2o)).

$$\min \sum_{i\in\mathscr{F}} \sum_{t\in\mathscr{T}_i^{\text{dep}}} g_{it}(w_{i,t-1}^{\text{dep}} - w_{i,t}^{\text{dep}}), \tag{11a}$$

$$\text{s.t. Scheduling constraints: Equations (1b)– (1f),} \tag{11b}$$

$$a_{kq} \sum_{i \in \mathscr{F}_k^{\text{dep}}} (w_{i,t-1}^{\text{dep}} - w_{it}^{\text{dep}}) + b_{kq} \sum_{i \in \mathscr{F}_k^{\text{arr}}} (w_{i,t-1}^{\text{arr}} - w_{it}^{\text{arr}}) \leq Q_{kq}^{\text{VMC}}, \forall q \in \mathscr{Q}_k,\ t \in \mathscr{T},\ k \in \mathscr{K}'. \tag{11c}$$

The results demonstrate the benefits of scale integration (i.e., capturing network-wide interdependencies) and scope integration (i.e., capturing interdependencies between scheduling and operations). Figure 4a shows that schedule optimization across larger networks results in greater expected delay reductions, for the same displacement "budget". For instance, displacing 1% of flights reduces expected delays by 20% in the 12-airport network but by nearly 30% in the 30-airport network. Figure 4b shows that our integrated approach dominates the sequential approach. As compared to the instance where all airports are subject to VMC-based schedule limits, our integrated approach achieves a Pareto improvement with larger expected delay reductions (38.69% vs. 35.24%) and a smaller displacement (2.4% vs. 3.8%). Obviously, the sequential approach can achieve a smaller displacement by applying schedule limits at a subset of airports but the expected delay reductions are then much smaller (e.g., 25% vs. 38.69% when approximately 2.5% of flights are displaced). This demonstrates the benefits of our modeling and computational approach, which optimizes scheduling interventions and MAGHO operations jointly across a network of airports.




(a) Different network sizes (b) Integrated vs. sequential approaches

**Figure 4 Optimal schedule displacement vs. MAGHO delays.**

Moreover, Figure 4a suggests that even moderate SI can result in significant operational benefits. For instance, rescheduling 1% of the flights across 6-30 airports reduces expected MAGHO delays by 20-30%. Moreover, each curve has a concave shape: significant delay savings can be achieved by rescheduling a small fraction of flights away from peak hours but returns are diminishing. These findings extent the insights from Pyrgiotis and Odoni (2016) and Jacquillat and Odoni (2015) to a

network setting. From a practical standpoint, this suggests opportunities to implement moderate SI, striking a middle ground between the mild demand management practices in place at US airports and strict schedule coordination in place outside the United States.

### 7.2. Spatial Patterns of SI and MAGHO

Figures 5 and 6 now show the role played by the different airports within the network. The figures plot the percentage of flights displaced and the expected delay reduction per airport, respectively, for different values of $\rho$. Their horizontal axes sort airports by non-increasing order of the number of flights displaced with $\rho = \rho_1$ (a proxy for demand-capacity imbalances).

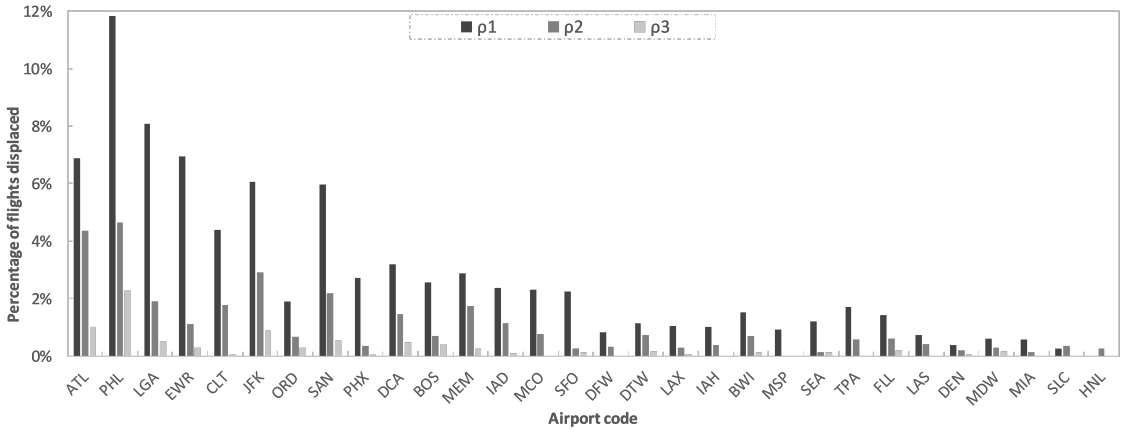



**Figure 5** **Percentage of flights displaced at each airport.**

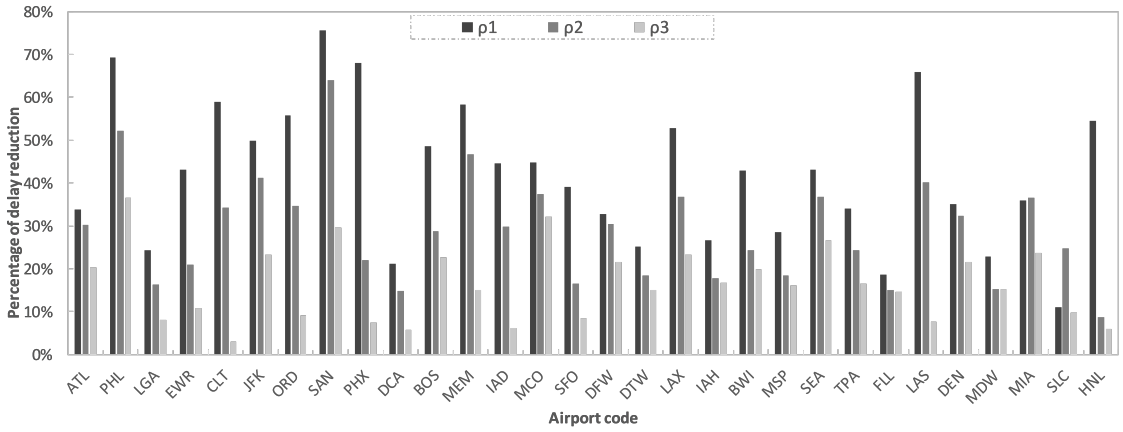



**Figure 6** **Percentage of expected delay reduction at each airport.**

Note that all airports are subject to SI, even the smallest ones. But displacement is not distributed evenly: for instance, busy airports such as ATL, PHL and LGA face large displacements. Delay reductions, in contrast, are much more evenly spread across airports. In fact, airports subject to the highest extents of displacement do not necessarily benefit the most operationally: for instance, LAX, LAS and MIA experience greater delay reductions, percent-wise, than ATL despite

facing a much smaller displacement. This suggests that optimal SI consider scheduling and capacity patterns at each airport as well as network-wide interdependencies. These insights contrast with existing practices in the United States, where scheduling interventions are applied at only a few of the busiest airports, and with schedule coordination outside the United States, where declared capacities are set at each airport independently based on local operating capacities alone.

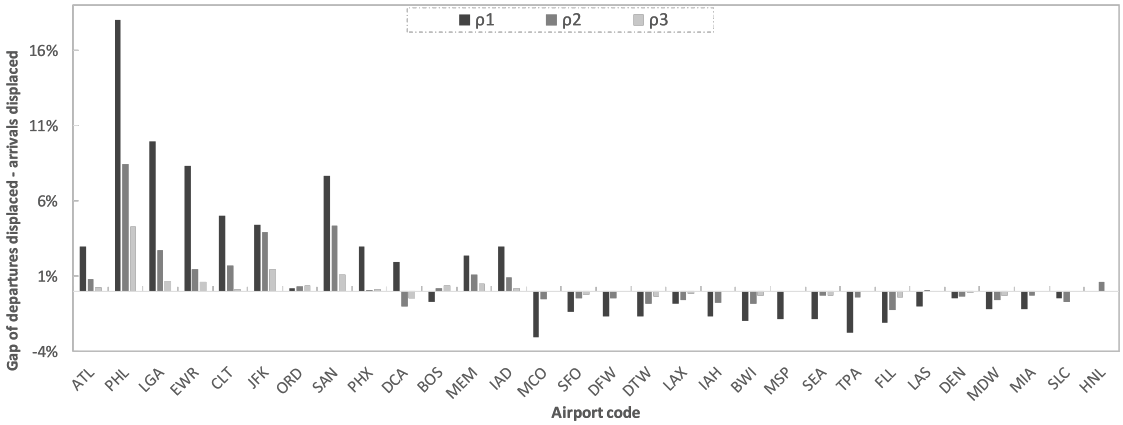



**Figure 7 Percentage difference between departures and arrivals displaced at each airport.**

Finally, Figure 7 plots the difference between the percentage of departures and arrivals displaced at each airport. It suggests that more departures than arrivals are displaced at busier airports (left-hand side)—to prevent the formation and propagation of delays at downstream airports. In contrast, more arrivals are displaced at less busy airports (right-hand side)—to absorb delays at nodes where downstream disruptions are less significant. This, again, underscores the impact of accounting for network-wide MAGHO operations in schedule optimization.

### 7.3. Temporal Patterns of SI and MAGHO

Last, we investigate patterns of scheduling displacement and flight delays over time of day. Table 8 shows the number of flights displaced to earlier or later periods and their average departure/arrival times. Overall, more flights are displaced to earlier periods than to later periods. In other words, SI leverages the opportunity to reschedule flights to earlier times for delay mitigation; in contrast, flights that need to be operated later than their requested times may be delayed in adverse scenarios through MAGHO rather than displaced *ex ante* through SI (especially for larger values of $\rho$). Moreover, flights displaced to earlier periods have earlier average departure/arrival times than flights displaced to later periods. In other words, flights are rescheduled at earlier times of the day to prevent the formation and propagation of delays across airport networks and over time; in contrast, ripple benefits of delay reductions are smaller at later time periods.

Figure 8 further illustrates the ripple effects of SI by plotting the percentage of delay reduction over the day. The main observation is that most significant delay reductions occur at earlier

**Table 8 Flights displaced to earlier or later periods.**

| | Flights displaced to earlier periods | | | Flights displaced to later periods | | |
|---|---|---|---|---|---|---|
| $\rho$ | Number | Avg. dep. time | Avg. arr. time | Number | Avg. dep. time | Avg. arr. time |
| $\rho_1$ | 407 | 10:57 am | 1:04 pm | 236 | 1:43 pm | 3:52 pm |
| $\rho_2$ | 280 | 10:12 am | 11:46 am | 0 | —— | —— |
| $\rho_3$ | 75 | 10:23 am | 11:54 am | 0 | —— | —— |

times. At later periods, the model simply trades off the costs of delays with the costs of schedule displacement—resulting in average delay reductions of 30% with $\rho = \rho_1$ and 10-20% with $\rho = \rho_2$ or $\rho = \rho_3$. But comparatively more flights are displaced in earlier periods, resulting in larger percent-wise delay reductions. In other words, SI are more aggressive in earlier periods to avoid the formation of delays that can propagate across the network of airports and over time.

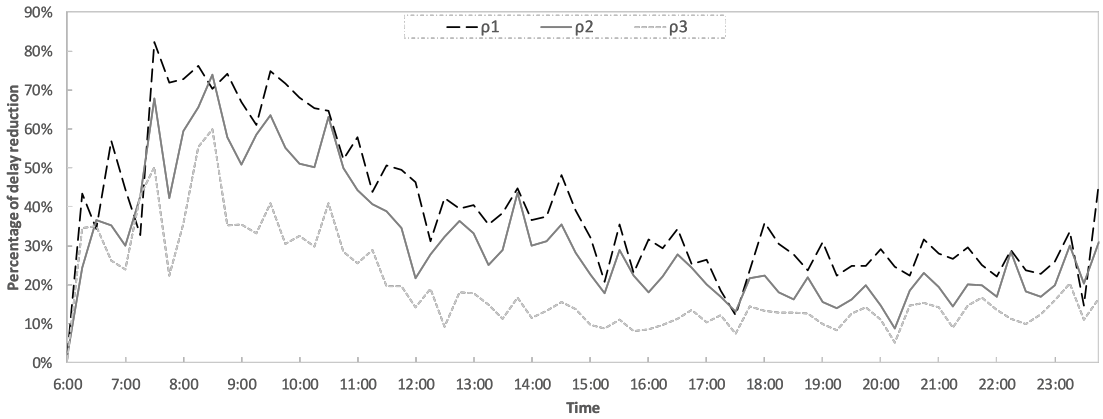



**Figure 8 Percentage of expected delay reduction over time of day.**

In summary, results demonstrate the benefits of scale and scope integration, and the potential of (IMSOAN) to support SI decisions regarding how many flights to displace, at which airports and at what times of day. Specifically, optimal scheduling interventions occur throughout the network and throughout the day, but are concentrated at the busiest airports and at earlier times of the day to minimize the formation and propagation of delays over space and time.

## 8. Conclusion

This paper has developed an *Integrated Model of Scheduling and Operations in Airport Networks* (IMSOAN), which jointly optimizes scheduling interventions at the strategic level (before uncertainty is resolved) and ground-holding operations at the tactical level (after uncertainty is resolved), in a network of airports. The model is formulated as a stochastic program with integer recourse. We have derived new *dual integer cuts* from the dual linear programmingrelaxation of the second-stage problem. These cuts provide a novel decomposition algorithm for two-stage stochastic programs

with integer recourse, as well as tighter optimality cuts than traditional Benders cuts for two-stage stochastic programs with continuous or integer recourse. The algorithm also incorporates original neighborhood constraints, which accelerate convergence by shifting from exploration to exploitation at later stages. Finally, we have proposed a data-driven scenario generation approach—cast as a $p$-median problem—to select scenarios from historical observations in stochastic programming.

The (IMSOAN) has been applied to the US National Airspace System. In our largest test instances with 8.2 million integer variables and 20.8 million constraints, direct CPLEX implementation does not even yield a feasible solution but the proposed algorithm provides near-optimal solutions in reasonable computational times. Results also demonstrate the benefits of our dual integer cuts (vs. traditional Benders cuts), of our neighborhood constraints (vs. existing acceleration strategies), of our stochastic optimization approach (vs. deterministic approaches) and of our scenario generation procedure (vs. heuristic procedures). Case study results show that our integrated network-wide approach to scheduling and operations yields significant benefits, as compared to partial approaches focused on a subset of the network, and to sequential approaches where scheduling and operating decisions are made separately. At schedule-coordinated airports, this approach can provide decision-making support to set declared capacities and allocate slots in a way that captures interdependencies across multiple airports and interdependencies between flight schedules and air traffic operations. At US airports, it can determine to which extent, where (i.e., at which airports) and when (i.e., at which times) to consider congestion-mitigating scheduling adjustments.

The work reported in this paper can be extended in several ways. First, this paper has considered the multi-airport ground-holding problem in the second stage. An important opportunity lies in integrating more complex air traffic flow management models with en-route capacity restrictions and routing decisions into the first-stage optimization of network-wide scheduling interventions. Second, the second-stage model could also be augmented to capture more granular airport operating decisions, such as the selection of runway configurations. Third, the first-stage model of scheduling interventions could be extended to capture complexities arising at schedule-coordinated airports (e.g., schedule regularity requirements across multiple days) and to design, optimize and assess market-based demand management mechanisms based on congestion pricing and slot auctions. Last, the positive results of our algorithm motivate its application to similarly-structured stochastic programs with integer or continuous recourse. Ultimately, this paper provides methodological foundations to enhance efficiency in various networked systems through tighter integration of strategic and tactical decisions under operating uncertainty.

## References

Ahmed S, Sahinidis NV (2003) An approximation scheme for stochastic integer programs arising in capacity expansion. *Operations Research* 51(3):461–471.

Ahmed S, Tawarmalani M, Sahinidis N (2004) A fnite branch and bound algorithm for two-stage stochastic integer programs. *Mathematical Programming* 100(2):355–377.

Angulo G, Ahmed S, Dey SS (2016) Improving the integer l-shaped method. *INFORMS Journal on Computing* 28(3):483–499.

Balakrishnan H, Chandran B (2010) Algorithms for Scheduling Runway Operations Under Constrained Position Shifting. *Operations Research* 58(6):1650–1665.

Baldacci R, Christofides N, Mingozzi A (2008) An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming* 115(2):351–385.

Baldacci R, Mingozzi A (2009) A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming* 120(2):347.

Ball M, Barnhart C, Dresner M, Hansen M, Neels K, Odoni A, Peterson E, Sherry L, Trani A, Zou B (2010) Total delay impact study. Technical report, National Center of Excellence for Aviation Operations Research, College Park, MD.

Ball M, Barnhart C, Nemhauser G, Odoni A (2007) Air transportation: Irregular operations and control. volume 14, 1–67 (Elsevier).

Ball M, Hoffman R, Odoni A, Rifkin R (2003) A stochastic integer program with dual network structure and its application to the ground-holding problem. *Operations Research* 51(1):167–171.

Barnhart C, Fearing D, Vaze V (2014) Modeling passenger travel and delays in the national air transportation system. *Operations Research* 62(3):580–601.

Benders J (1962) Partitioning procedures for solving mixed-variable programming problems. *Numerische Mathematic* 4(1):238–252.

Bertsimas D, Lulli G, Odoni A (2011) An integer optimization approach to large-scale air traffic flow management. *Operations Research* 59(1):211–227.

Bertsimas D, Stock Patterson S (1998) The air traffic flow management problem with enroute capacities. *Operations Research* 46(3):406–422.

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

Carlin A, Park P (1970) Marginal cost pricing of airport runway capacity. *American Economic Review* 60(3):310–319.

Caroe C, Schultz R (1997) Dual decomposition in stochastic integer programming. *Operations Research Letters* 24(1-2):37–45.

Caroe C, Schultz R (1999) L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming* 83(1-3):451–464.

Carrión M, Philpott AB, Conejo AJ, Arroyo JM (2007) A stochastic programming approach to electric energy procurement for large consumers. *IEEE Transactions on Power Systems* 22(2):744–754.

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

Czerny A, Forsyth P, Gillen D, Niemeier H (2008) *Airport Slots* (Ashgate Publishing Ltd).

de Neufville R, Odoni A (2013) *Airport Systems: Planning, Design and Management* (McGraw-Hill), 2nd edition.

Dupačová J, Gröwe-Kuska N, Römisch W (2003) Scenario reduction in stochastic programming. *Mathematical Programming* 95(3):493–511.

Federal Aviation Administration (2004) Airport capacity benchmark report. Technical report.

Federal Aviation Administration (2017) Aviation System Performance Metrics (ASPM) database. Accessed February 1, 2017. Available at: https://aspm.faa.gov/apm/sys/main.asp.

Fischetti M, Lodi A (2003) Local branching. *Mathematical Programming* 98(1-3):23–47.

Fischetti M, Toth P (1988) An additive approach for the optimal solution of the prize collecting traveling salesman problem. *Vehicle routing: Methods and studies* 231:319–343.

Fischetti M, Toth P (1989) An additive bounding procedure for combinatorial optimization problems. *Operations Research* 37(2):319–328.

Fischetti M, Toth P (1992) An additive bounding procedure for the asymmetric travelling salesman problem. *Mathematical Programming* 53(1–3):173–197.

Fischetti M, Toth P (1993) An efficient algorithm for the min-sum arborescence problem on complete digraphs. *ORSA Journal on Computing* 5(4):426–434.

Gilbo E (1993) Airport capacity: Representation, estimation, optimization. *IEEE Transactions on Control Systems Technology* 1(3):144–154.

Gillen D, Jacquillat A, Odoni A (2016) Airport demand management: The operations research and economics perspectives and potential synergies. *Transportation Research Part A: Policy and Practice* 94(3):495–513.

Gunpinar S, Centeno G (2015) Stochastic integer programming models for reducing wastages and shortages of blood products at hospitals. *Computers & Operations Research* 54:129–141.

Jacquillat A, Odoni A (2015) An integrated scheduling and operations approach to airport congestion mitigation. *Operations Research* 63(6):1390–1410.

Jacquillat A, Vaze V (2018) Interairline equity in airport scheduling interventions. *Transportation Science* 52(4):739–1034.

Jiang Y, Zografos K (2018) A bi-objective efficiency-fairness model for scheduling slots at congested airports. *Working paper* .

Jones J, Lovell D, Ball M (2018) Stochastic optimization models for transferring delay along flight trajectories to reduce fuel usage. *Transportation Science* 52(1):1–227.

Kim K, Mehrotra S (2015) A two-stage stochastic integer programming approach to integrated staffing and scheduling with application to nurse management. *Operations Research* 63(6):1431–1451.

Kleywegt A, Shapiro A, Homem-De-Mello T (2001) The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12(2):479–502.

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

Laporte G, Louveaux FV, Van Hamme L (2002) An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research* 50(3):415–423.

Louveaux FV, Schultz R (2003) Stochastic integer programming. volume 10, 213–266 (Elsevier).

Magnanti TL, Wong RT (1981) Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research* 29(3):464–484.

Morales JM, Pineda S, Conejo AJ, Carrion M (2009) Scenario reduction for futures market trading in electricity markets. *IEEE Transactions on Power Systems* 24(2):878–888.

Mukherjee A, Hansen M (2007) A dynamic stochastic model for the single airport ground holding problem. *Transportation Science* 41(4):444–456.

Ntaimo L (2010) Disjunctive decomposition for two-stage stochastic mixed-binary programs with random recourse. *Operations research* 58(1):229–243.

Odoni A (1987) *Flow Control of Congested Networks*, chapter The Flow Management Problem in Air Traffic Control (Springer-Verlag).

Pellegrini P, Bolic T, Castelli L, Resenti R (2017) SOSTA: An effective model for the simultaneous optimisation of airport slot allocation. *Transportation Research Part E: Logistics and Transportation Review* 99:34–53.

Pellegrini P, Castelli L, Pesenti R (2012) Metaheuristic algorithms for the simultaneous slot allocation problem. *IET Intelligent Transport Systems* 6(4):453–462.

Pyrgiotis N (2011) *A Stochastic and Dynamic Model of Delay Propagation Within an Airport Network For Policy Analysis*. Ph.D. thesis, Massachusetts Institute of Technology.

Pyrgiotis N, Odoni A (2016) On the impact of scheduling limits: A case study at newark international airport. *Transportation Science* 50(1):150–165.

Rassenti S, Smith V, Bulfin R (1982) A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics* 13(2):402–417.

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

Rei W, Cordeau JF, Gendreau M, Soriano P (2009) Accelerating benders decomposition by local branching. *INFORMS Journal on Computing* 21(2):333–345.

Ribeiro N, Jacquillat A, Antunes A (2019) A large-scale neighborhood search approach to airport slot allocation. *Transportation Science* in press.

Ribeiro N, Jacquillat A, Antunes A, Odoni A, Pita J (2017) An optimization approach for airport slot allocation under IATA guidelines. *Transportation Research Part B: Methodological* 112:132–156.

Richetta O, Odoni A (1993) Solving optimally the static ground-holding policy problem in air traffic control. *Transportation Science* 27(3):228–237.

Römisch W (2009) Scenario reduction techniques in stochastic programming. *International Symposium on Stochastic Algorithms*, 1–14 (Springer).

Schultz R (1993) Continuity properties of expectation functions in stochastic integer programming. *Mathematics of Operations Research* 18(3):578–589.

Schultz R, Stougie L, van der Vlerk M (1998) Solving stochastic programs with integer recourse by enumeration: A framework using grobner basis reductions. *Mathematical Programming* 83(1-3):229–252.

Sen S (2005) Algorithms for stochastic mixed-integer programming models. volume 12, 515–558 (Elsevier).

Sen S, Higle J (2005) The $c^3$ theorem and a $d^2$ algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming* 104(1):1–20.

Sen S, Sherali H (2006) Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming* 106(2):203–223.

Sherali H, Lunday B (2013) On generating maximal nondominated benders cuts. *Annals of Operations Research* 210(1):57–72.

Simaiakis I (2012) *Analysis, Modeling and Control of the Airport Departure Process*. Ph.D. thesis, Massachusetts Institute of Technology.

Solveling G, Solak S, Clarke JP, Johnson E (2011) Runway Operations Optimization in the Presence of Uncertainties. *AIAA Journal of Guidance, Control and Dynamics* 34(5):1373–1382.

Terrab M, Odoni A (1993) Strategic flow management for air traffic control. *Operations Research* 41(1):138–152.

Vossen TW, Hoffman R, Mukherjee A (2012) Air traffic flow management. *Quantitative problem solving methods in the airline industry*, 385–453 (Springer).

Vranas P, Bertsimas D, Odoni A (1994) The multi-airport ground-holding problem in air traffic control. *Operations Research* 42(2):249–261.

Zhang M, Kucukyavuz S (2014) Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs. *SIAM Journal on Optimization* 24(4):1933–1951.

Zografos K, Androutsopoulos K, Madas M (2017) Minding the gap: Optimizing airport schedule displacement and acceptability. *Transportation Research Part A: Policy and Practice* 114:203–221.

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

Zou J, Ahmed S, Sun XA (2017) Stochastic dual dynamic integer programming. *Mathematical Programming* 1–42.

## Appendix A: Appendix to Model Formulation (Section 3)

### A.1. Proof of that Equations (2l) and (2m) are Valid Inequalities

Let us consider a given scenario $s \in \mathscr{S}$ and a given flight $i \in \mathscr{F}$. We know from Equation (2d) that:

$$\sum_{t \in \mathscr{T}} x^{\text{dep}}_{its} \geq \sum_{t \in \mathscr{T}} w^{\text{dep}}_{it}. \tag{A.1}$$

This implies that $x^{\text{dep}}_{its} \geq w^{\text{dep}}_{it}$ for all $t \in \mathscr{T}$. Indeed, let us assume by contradiction that $x^{\text{dep}}_{i\tau s} < w^{\text{dep}}_{i\tau}$ for some $\tau \in \mathscr{T}$. Then we know from Equations (1b) and (2b) that $w^{\text{dep}}_{it} = 1$ for all $t \leq \tau$ and $x^{\text{dep}}_{its} = 0$ for all $t \geq \tau$. This implies that:

$$\sum_{t \in \mathscr{T}} x^{\text{dep}}_{its} = \sum_{t < \tau} x^{\text{dep}}_{its} < \sum_{t \leq \tau} w^{\text{dep}}_{it} \leq \sum_{t \in \mathscr{T}} w^{\text{dep}}_{it},$$

which contradicts Equation (A.1). This proves Equation (2l). Note that Equation (2l) is only associated with $t \in \mathscr{T}^{\text{dep}}_i$. The fact that $x^{\text{dep}}_{its} \geq w^{\text{dep}}_{it}$ for all $t \in \mathscr{T} \setminus \mathscr{T}^{\text{dep}}_i$ stems directly from the definition of our variables.

We now turn to Equation (2m). We know that, from Equations (2i) and (A.1):

$$\sum_{t \in \mathscr{T}} x^{\text{arr}}_{its} \geq \sum_{t \in \mathscr{T}} x^{\text{dep}}_{its} + \Delta^{\text{min}}_i \geq \sum_{t \in \mathscr{T}} w^{\text{dep}}_{it} + \Delta^{\text{min}}_i. \tag{A.2}$$

Moreover, we have from Equation (1d):

$$\sum_{t \in \mathscr{T}} w^{\text{arr}}_{it} = \sum_{t \in \mathscr{T}} w^{\text{dep}}_{it} + \Delta^{\text{sch}}_i. \tag{A.3}$$

Therefore, from Equations (A.2) and (A.3), we obtain:

$$\sum_{t \in \mathscr{T}} x^{\text{arr}}_{its} \geq \sum_{t \in \mathscr{T}} w^{\text{arr}}_{it} - \left(\Delta^{\text{sch}}_i - \Delta^{\text{min}}_i\right). \tag{A.4}$$

Recall that $x^{\text{arr}}_{i\bullet s}$ and $w^{\text{arr}}_{i\bullet}$ are of the form $(1, \cdots, 1, 0, \cdots, 0)$, and thus Equation (A.4) implies that $x^{\text{arr}}_{i,t-\left(\Delta^{\text{sch}}_i - \Delta^{\text{min}}_i\right),s} \geq w^{\text{arr}}_{it}$ for all $t \in \mathscr{T}$. This proves Equation (2m). Again, the fact that $x^{\text{arr}}_{i,t-\left(\Delta^{\text{sch}}_i - \Delta^{\text{min}}_i\right),s} \geq w^{\text{arr}}_{it}$ for all $t \in \mathscr{T} \setminus \mathscr{T}^{\text{arr}}_i$ stems directly from the definition of our variables. □

### A.2. Effectiveness of Valid Inequalities (2l) and (2m)

Consider an instance with $T = 4$, $F = 2$, $\delta = 1$ and $S = 1$. Suppose $S^{\text{dep}}_1 = S^{\text{dep}}_2 = 2$ and both flights depart from the same origin airport. We consider a feasible scheduling interventions solution $w^{\text{dep}}_{1\bullet} = w^{\text{dep}}_{2\bullet} = (1, 1, 0, 0)$ (i.e., none of the two flights is displaced). Suppose the airport's departure capacity is equal to 1 for all $t$ under scenario 1. Then, $x^{\text{dep}}_{1,\bullet,1} = x^{\text{dep}}_{2,\bullet,1} = (1, 0.5, 0.5, 0)$ is a feasible solution of the linear programming relaxation of the second-stage model if Constraints (2l) and (2m) are omitted. However, after adding valid inequalities (2l) and (2m), this solution becomes infeasible. In this case, the inclusion of the valid constraints leads to the integer solution $x^{\text{dep}}_{1,\bullet,1} = (1, 1, 0, 0)$ and $x^{\text{dep}}_{2,\bullet,1} = (1, 1, 1, 0)$. In the general case, these valid constraints do not eliminate all fractional solutions, but considerably tighten the formulation of (IMSOAN).

## Appendix B: Appendix to Algorithm Development (Section 4)

### B.1. Formulation of $\left(\mathrm{D}-\overline{\mathrm{SP}}\right)$

We introduce the following dual variables. For simplicity, we omit the superscript $s$.

- $\boldsymbol{\mu} = (\{\mu_{it}^{\text{dep}} \leq 0 |\ i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{dep}}\}, \{\mu_{it}^{\text{arr}} \leq 0 |\ i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{arr}}\})$ are the dual variables associated with Equations (2b) and (2c).
- $\boldsymbol{\lambda} = (\{\lambda_i^{\text{dep}} \in \mathbb{R} |\ i \in \mathscr{F}\},\ \{\lambda_i^{\text{arr}} \leq 0 |\ i \in \mathscr{F}\})$ are the dual variables associated with Equations (2d) and (2e).
- $\boldsymbol{\eta} = (\{\eta_i^{\text{dep}} \leq 0 |\ i \in \mathscr{F}\},\ \{\eta_i^{\text{arr}} \leq 0 |\ i \in \mathscr{F}\})$ are the dual variables associated with Equations (2f) and (2g).
- $\boldsymbol{\xi} = \{\xi_{ij} \geq 0 |\ (i,j) \in \mathscr{C}^2\}$ are the dual variables associated with Equation (2h).
- $\boldsymbol{\alpha} = \{\alpha_i \geq 0 |\ i \in \mathscr{F}\}$ are the dual variables associated with Equation (2i).
- $\boldsymbol{\beta} = \{\beta_i \leq 0 |\ i \in \mathscr{F}\}$ are the dual variables associated with Equation (2j).
- $\boldsymbol{\chi} = \{\chi_{ktq} \leq 0 |\ q \in \mathscr{Q}_k,\ t \in \mathscr{T},\ k \in \mathscr{K}\}$ are the dual variables associated with Equation (2k).
- $\boldsymbol{\nu} = (\{\nu_{it}^{\text{dep}} \geq 0 |\ i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{dep}}\}, \{\nu_{it}^{\text{arr}} \geq 0 |\ i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{arr}},\ t-(\Delta_i^{\text{sch}}-\Delta_i^{\text{min}}) \geq \min(\mathscr{T}_i^{\text{arr}})\})$ are the dual variables associated with Equations (2l) and (2m).

The $\left(\mathrm{D}-\overline{\mathrm{SP}}\right)$ formulation is as follows:

$$\begin{aligned}
\max\ & \sum_{i\in\mathscr{F}} \mu_{i,S_i^{\text{dep}}-\delta+1}^{\text{dep}} + \sum_{i\in\mathscr{F}} \mu_{i,S_i^{\text{dep}}-\delta+\Delta_i^{\text{min}}+1}^{\text{arr}} + \sum_{i\in\mathscr{F}} \lambda_i^{\text{dep}} \Big(\sum_{t\in\mathscr{T}} w_{it}^{\text{dep}} - \sum_{t\in\mathscr{T}\setminus\mathscr{T}_i^{\text{dep}}} x_{its}^{\text{dep}}\Big) \\
& + \sum_{i\in\mathscr{F}} \lambda_i^{\text{arr}} \Big(\sum_{t\in\mathscr{T}} w_{it}^{\text{arr}} - \sum_{t\in\mathscr{T}\setminus\mathscr{T}_i^{\text{arr}}} x_{its}^{\text{arr}}\Big) + \sum_{i\in\mathscr{F}} \eta_i^{\text{dep}} l_i^{\text{dep}} + \sum_{i\in\mathscr{F}} \eta_i^{\text{arr}} l_i^{\text{arr}} \\
& + \sum_{(i,j)\in\mathscr{C}^2} \xi_{ij} \Big(\tau_{ij}^{(2)} - \sum_{t\in\mathscr{T}\setminus\mathscr{T}_j^{\text{dep}}} x_{jts}^{\text{dep}} + \sum_{t\in\mathscr{T}\setminus\mathscr{T}_i^{\text{arr}}} x_{its}^{\text{arr}}\Big) \\
& + \sum_{i\in\mathscr{F}} \alpha_i \Big(\Delta_i^{\text{min}} - \sum_{t\in\mathscr{T}\setminus\mathscr{T}_i^{\text{arr}}} x_{its}^{\text{arr}} + \sum_{t\in\mathscr{T}\setminus\mathscr{T}_i^{\text{dep}}} x_{its}^{\text{dep}}\Big) + \sum_{i\in\mathscr{F}} \beta_i \Big(\Delta_i^{\text{max}} - \sum_{t\in\mathscr{T}\setminus\mathscr{T}_i^{\text{arr}}} x_{its}^{\text{arr}} + \sum_{t\in\mathscr{T}\setminus\mathscr{T}_i^{\text{dep}}} x_{its}^{\text{dep}}\Big) \\
& + \sum_{k\in\mathscr{K}} \sum_{t\in\mathscr{T}} \sum_{q\in\mathscr{Q}_k} \chi_{ktq} \Big(Q_{kq}(\phi_{kts}) - \sum_{i\in\mathscr{F}_k^{\text{dep}} | S_i^{\text{dep}}-\delta+1=t} a_{kq} - \sum_{i\in\mathscr{F}_k^{\text{arr}} | S_i^{\text{dep}}-\delta+\Delta_i^{\text{min}}+1=t} b_{kq}\Big) \\
& + \sum_{i\in\mathscr{F}} \sum_{t\in\mathscr{T}_i^{\text{dep}}} \nu_{it}^{\text{dep}} w_{it}^{\text{dep}} + \sum_{i\in\mathscr{F}} \sum_{t\in\mathscr{T}_i^{\text{arr}}} \nu_{it}^{\text{arr}} w_{it}^{\text{arr}}, \qquad \text{(B.1a)} \\
\text{s.t.}\ & \mu_{it}^{\text{dep}} - \mu_{i,t+1}^{\text{dep}} + \lambda_i^{\text{dep}} + \sum_{(j,i)\in\mathscr{C}^2} \xi_{ji} - \alpha_i - \beta_i + \sum_{q\in\mathscr{Q}_k} a_{kq} \left(\chi_{k,t+1,q} - \chi_{ktq}\right) + \nu_{it}^{\text{dep}} \leq 0 \\
& \qquad \forall i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{dep}}, \qquad \text{(B.1b)} \\
& \mu_{it}^{\text{arr}} - \mu_{i,t+1}^{\text{arr}} + \lambda_i^{\text{arr}} - \sum_{(i,j)\in\mathscr{C}^2} \xi_{ij} + \alpha_i + \beta_i + \sum_{q\in\mathscr{Q}_k} b_{kq} \left(\chi_{k,t+1,q} - \chi_{ktq}\right) + \nu_{i,t+(\Delta_i^{\text{sch}}-\Delta_i^{\text{min}})}^{\text{arr}} \leq 0 \\
& \qquad \forall i \in \mathscr{F},\ t \in \mathscr{T}_i^{\text{arr}}, \qquad \text{(B.1c)} \\
& -\lambda_i^{\text{dep}} + \eta_i^{\text{dep}} \leq c_i^{\text{dep}} \quad \forall i \in \mathscr{F}, \qquad \text{(B.1d)} \\
& -\lambda_i^{\text{arr}} + \eta_i^{\text{arr}} \leq c_i^{\text{arr}} \quad \forall i \in \mathscr{F}, \qquad \text{(B.1e)} \\
& \mu_{it}^{\text{dep}}, \mu_{it}^{\text{arr}} \in \boldsymbol{\mu}, \lambda_i^{\text{dep}}, \lambda_i^{\text{arr}} \in \boldsymbol{\lambda}, \eta_i^{\text{dep}}, \eta_i^{\text{arr}} \in \boldsymbol{\eta}, \xi_{ij} \in \boldsymbol{\xi}, \alpha_i \in \boldsymbol{\alpha}, \beta_i \in \boldsymbol{\beta}, \chi_{ktq} \in \boldsymbol{\chi}, \nu_{it}^{\text{dep}}, \nu_{it}^{\text{arr}} \in \boldsymbol{\nu}. \qquad \text{(B.1f)}
\end{aligned}$$

Note that for all $x_{its}^{\text{dep}}$ is a parameter in $\left(\overline{\text{SP}}\right)$ for $t \in \mathscr{T} \setminus \mathscr{T}_i^{\text{dep}}$, and so is $x_{its}^{\text{arr}}$ for $t \in \mathscr{T} \setminus \mathscr{T}_i^{\text{arr}}$. Hence, they are included in the dual objective function (B.1a). Equations (B.1b), (B.1c), (B.1d) and (B.1e) are the dual constraints corresponding to the decision variables $\boldsymbol{x}_s^{\text{dep}}$, $\boldsymbol{x}_s^{\text{arr}}$, $\boldsymbol{v}_s^{\text{dep}}$ and $\boldsymbol{v}_s^{\text{arr}}$ of $\left(\overline{\text{SP}}\right)$, respectively, in scenario $s \in \mathscr{S}$. Let $\boldsymbol{\Pi}_s = (\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\chi}, \boldsymbol{\nu})$ denote a feasible dual solution of $\left(\text{D} - \overline{\text{SP}}\right)$.

### B.2. Proof of Proposition 1

To prove Proposition 1, we make use of the following lemma:

LEMMA 1. *Consider an integer program in standard form* $z(\text{F}) = \min\{\boldsymbol{cx} | s.t.\ \boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{x} \in \mathbb{Z}_+^n\}$ *with* $n$ *integer variables and* $m$ *constraints. Let* $\boldsymbol{\pi}$ *be a feasible solution to the dual linear programming relaxation of (F). For all* $i \in \{1, \cdots, n\}$*, let* $\gamma_i'$ *be the reduced cost of variable* $x_i$ *associated with* $\boldsymbol{\pi}$*. The optimal solution of (F), denoted by* $\boldsymbol{x}^* = \{x_i^*,\ i = 1, \cdots, n\}$*, satisfies*

$$z(\text{F}) = \boldsymbol{\pi b} + \sum_{i=1}^{n} \gamma_i' x_i^*. \tag{B.2}$$

PROOF. Let $\boldsymbol{\pi}$ be a feasible solution to the dual linear programming relaxation of (F). $\boldsymbol{\pi b}$ is the objective function of the dual problem. We denote by $\boldsymbol{a}_i$ the $i^{th}$ column of the matrix $\boldsymbol{A}$. Then the reduced cost for each column $i \in \{1, \cdots, n\}$ (or each variable $x_i$) is $\gamma_i' = c_i - \boldsymbol{\pi a}_i$, where $c_i$ is the $i^{\text{th}}$ coefficient of $\boldsymbol{c}$. We have:

$$\begin{aligned} \boldsymbol{\pi b} + \sum_{i=1}^{n} \gamma_i' x_i^* &= \boldsymbol{\pi b} + \sum_{i=1}^{n} (c_i - \boldsymbol{\pi a}_i) x_i^* \\ &= \underbrace{\sum_{i=1}^{n} c_i x_i^*}_{=z(\text{F})} + \boldsymbol{\pi} \underbrace{\left( \boldsymbol{b} - \sum_{i=1}^{n} \boldsymbol{a}_i x_i^* \right)}_{=0}. \quad \square \end{aligned}$$

Although Lemma 1 is proved for a standard formulation with constraints $\boldsymbol{Ax} = \boldsymbol{b}$, it also holds for any non-standard formulation with constraints $\boldsymbol{Ax} \leq \boldsymbol{b}$. This is obtained by adding slack variables, which does not change its dual linear programming relaxation. It is worth mentioning that Lemma 1 is also a starting point of the additive bounding procedure from Fischetti and Toth (1989), but it is applied in a very different way in this paper. Next, we can prove Proposition 1.

PROOF OF PROPOSITION 1. Let us consider a scenario $s \in \mathscr{S}$. Let $\boldsymbol{w}$ be a feasible solution to (MP) and $\boldsymbol{\Pi}_s$ be a feasible solution to the $\left(\text{D} - \overline{\text{SP}}\right)$ problem. Note that the feasible region of the $\left(\text{D} - \overline{\text{SP}}\right)$ problem does not depend on the chosen (MP) solution $\boldsymbol{w}$ (Equations (B.1b) to (B.1f)). Then, with the slack variables $\boldsymbol{y}_s$, the following equation holds from Lemma 1:

$$\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s) = \boldsymbol{\Pi}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \boldsymbol{x}_s^* \gamma'(\boldsymbol{x}_s) + \boldsymbol{v}_s^* \gamma'(\boldsymbol{v}_s) + \boldsymbol{y}_s^* \gamma'(\boldsymbol{y}_s), \tag{B.3}$$

where $(\boldsymbol{x}_s^*, \boldsymbol{v}_s^*, \boldsymbol{y}_s^*)$ denotes the optimal IP solution of (SP) with respect to the feasible (MP) solution $\boldsymbol{w}$. $\square$

### B.3. Remark on Equation (5)

In Equation (5), the second-stage objective $\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s)$ can be mainly approximated by the dual objective and the term in $\boldsymbol{x}_s$ (i.e., the first two terms of the right-hand side). The remaining two terms can thus be ignored without considerably deteriorating the quality of the lower bound.

First, the term in $\boldsymbol{v}_s$ can be removed by adjusting the feasible dual solution $\boldsymbol{\Pi}_s$. Given any feasible dual solution $\boldsymbol{\Pi}_s = (\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\chi}, \boldsymbol{\nu})$ of $\left(\mathrm{D}-\overline{\mathrm{SP}}\right)$ in scenario $s$, it must hold that $\lambda_i^{\mathrm{dep}} \geq \eta_i^{\mathrm{dep}} - c_i^{\mathrm{dep}}$ and $\lambda_i^{\mathrm{arr}} \geq \eta_i^{\mathrm{arr}} - c_i^{\mathrm{arr}}$ for $i \in \mathscr{F}$ based on Equations (B.1d) and (B.1e), respectively. Then, from $\boldsymbol{\Pi}_s$, one can always construct another feasible dual solution $\widehat{\boldsymbol{\Pi}}_s = (\boldsymbol{\mu}, \widehat{\boldsymbol{\lambda}}, \boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\chi}, \boldsymbol{\nu})$ by letting $\widehat{\lambda}_i^{\mathrm{dep}} = \eta_i^{\mathrm{dep}} - c_i^{\mathrm{dep}} \leq \lambda_i^{\mathrm{dep}}$ and $\widehat{\lambda}_i^{\mathrm{arr}} = \eta_i^{\mathrm{arr}} - c_i^{\mathrm{arr}} \leq \lambda_i^{\mathrm{arr}}$ for all $i \in \mathscr{F}$. It is clear that $\widehat{\boldsymbol{\Pi}}_s$ is feasible to $\left(\mathrm{D}-\overline{\mathrm{SP}}\right)$. According to Lemma 1, it holds that $\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s) = \widehat{\boldsymbol{\Pi}}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \boldsymbol{x}_s^* \widehat{\gamma}'(\boldsymbol{x}_s) + \boldsymbol{v}_s^* \widehat{\gamma}'(\boldsymbol{v}_s) + \boldsymbol{y}_s^* \widehat{\gamma}'(\boldsymbol{y}_s)$, where $\widehat{\gamma}'(\cdot) \geq 0$ denotes the reduced cost of corresponding variables associated with $\widehat{\boldsymbol{\Pi}}_s$. Since $\widehat{\lambda}_i^{\mathrm{dep}} = \eta_i^{\mathrm{dep}} - c_i^{\mathrm{dep}}$ and $\widehat{\lambda}_i^{\mathrm{arr}} = \eta_i^{\mathrm{arr}} - c_i^{\mathrm{arr}}$ for all $i \in \mathscr{F}$, the reduced cost of variables $\boldsymbol{v}_s$ must be equal to zero (i.e., $\widehat{\gamma}'(\boldsymbol{v}_s) = \boldsymbol{0}$). Therefore, we have:

$$\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s) = \widehat{\boldsymbol{\Pi}}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \boldsymbol{x}_s^* \widehat{\gamma}'(\boldsymbol{x}_s) + \boldsymbol{y}_s^* \widehat{\gamma}'(\boldsymbol{y}_s), \tag{B.4}$$

In other words, the term in $\boldsymbol{v}_s$ (and its corresponding reduced cost) is not critical to evaluate $\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s)$—mainly due to the fact that $\boldsymbol{v}_s$ are auxiliary variables for (SP).

Second, $\boldsymbol{y}_s^* \gamma'(\boldsymbol{y}_s)$ in Equation (5) (or $\boldsymbol{y}_s^* \widehat{\gamma}'(\boldsymbol{y}_s)$ in Equation (B.4)) is equal to zero in general. To see this, let $\pi_{js} \leq 0$ be the dual variable of the $j^{th}$ constraint of $\left(\overline{\mathrm{SP}}\right)$ in scenario $s$ (with constraints of the form $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$). Then the reduced cost of the slack variable $y_{js}$ $(\geq 0)$ is $\gamma'(y_{js}) = -\pi_{js}$. We denote by $\bar{\boldsymbol{y}}_s^*$ the slack variable of the $\left(\overline{\mathrm{SP}}\right)$ at its LP optimum. If $y_{js}^* > 0$ (i.e., the $j^{\mathrm{th}}$ constraint is not binding in (SP)), it is likely that $\bar{y}_{js}^* > 0$ (i.e., the $j^{\mathrm{th}}$ constraint is not binding in $\left(\overline{\mathrm{SP}}\right)$ either). If $\bar{y}_{js}^* > 0$, we have $\pi_{js} = 0$ and $\gamma'(y_{js}) = 0$, such that $y_{js}^* \gamma'(y_{js}) = 0$. If $y_{js}^* = 0$, it is clear that $y_{js}^* \gamma'(y_{js}) = 0$.

This discussion indicates that the dual bound and the term $\boldsymbol{x}_s^* \widehat{\gamma}'(\boldsymbol{x}_s)$ capture the main primal and dual information to evaluate $\Psi(\boldsymbol{w}, \boldsymbol{\phi}_s)$. As a result, our approach in Theorem 1, which focuses on the first two terms of the right-hand side in Equation (5) provides a tight lower bound.

### B.4. Proof of Theorem 1

Let us consider a scenario $s \in \mathscr{S}$, a feasible (MP) solution $\boldsymbol{w}$ and a feasible solution $\boldsymbol{\Pi}_s$ of $\left(\mathrm{D}-\overline{\mathrm{SP}}\right)$. We adopt the convention that $\gamma'\left(x_{it's}^{\mathrm{arr}}\right) = 0$ for all $t' \notin \mathscr{T}_i^{\mathrm{arr}}$ and for all $i \in \mathscr{F}$. We have:

$$\begin{aligned}
\theta_s &\geq \Psi(\boldsymbol{w}, \boldsymbol{\phi}_s) \quad \text{(Equation (4b))} && \text{(B.5)}\\
&\geq \boldsymbol{\Pi}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \boldsymbol{x}_s^* \gamma'(\boldsymbol{x}_s) + \boldsymbol{v}_s^* \gamma'(\boldsymbol{v}_s) + \boldsymbol{y}_s^* \gamma'(\boldsymbol{y}_s) \quad \text{(Proposition 1)}\\
&\geq \boldsymbol{\Pi}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \boldsymbol{x}_s^* \gamma'(\boldsymbol{x}_s) \quad \text{because } \boldsymbol{v}_s^* \gamma'(\boldsymbol{v}_s) + \boldsymbol{y}_s^* \gamma'(\boldsymbol{y}_s) \geq 0\\
&= \boldsymbol{\Pi}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\mathrm{dep}}} x_{its}^{\mathrm{dep}*} \gamma'\left(x_{its}^{\mathrm{dep}}\right) + \sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\mathrm{arr}}} x_{its}^{\mathrm{arr}*} \gamma'\left(x_{its}^{\mathrm{arr}}\right)\\
&\geq \boldsymbol{\Pi}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\mathrm{dep}}} w_{it}^{\mathrm{dep}} \gamma'\left(x_{its}^{\mathrm{dep}}\right)\\
&\quad + \sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\mathrm{arr}}} w_{it}^{\mathrm{arr}} \gamma'\left(x_{i,t-\left(\Delta_{in}^{\mathrm{sch}} - \Delta_{in}^{\mathrm{min}}\right),s}^{\mathrm{arr}}\right) \quad \text{(Equations (2l) and (2m))}. && \text{(B.6)}
\end{aligned}$$

This proves the validity of our *dual integer cut*, in each scenario $s$:

$$\theta_s \geq \boldsymbol{\Pi}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\mathrm{dep}}} w_{it}^{\mathrm{dep}} \gamma'\left(x_{its}^{\mathrm{dep}}\right) + \sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\mathrm{arr}}} w_{it}^{\mathrm{arr}} \gamma'\left(x_{i,t-\left(\Delta_{in}^{\mathrm{sch}} - \Delta_{in}^{\mathrm{min}}\right),s}^{\mathrm{arr}}\right). \tag{B.7}$$

It is worth mentioning that if we start from Equation (B.4) (which also holds for any SI solution $\boldsymbol{w}$), we can derive the following cut: $\theta_s \geq \widehat{\boldsymbol{\Pi}}_s \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \sum_{i \in \mathscr{F}} \sum_{t \in \mathscr{T}_i^{\mathrm{dep}}} w_{it}^{\mathrm{dep}} \widehat{\gamma}'\left(x_{its}^{\mathrm{dep}}\right) +$

$\sum_{i\in\mathscr{F}}\sum_{t\in\mathscr{T}_i^{\text{arr}}} w_{it}^{\text{arr}}\widehat{\gamma}'\left(x^{\text{arr}}_{i,t-\left(\Delta_{in}^{\text{sch}}-\Delta_{in}^{\min}\right),s}\right)$, which is equivalent to Equation (B.7) (see Appendix B.3). This underscores that removing the decision variables $\boldsymbol{v}_s$ and their corresponding reduced cost (as in Equation (B.4)) will not deteriorate the tightness of our cut. In other words, the dual objective and the decision variables $\boldsymbol{x}_s$ capture the tightness of the cut. □

### B.5. Additional Details on Feasibility Cuts

Recall that a feasibility cut may be required: (i) at each iteration of the algorithm, if $\left(\overline{\text{SP}}\right)$ admits no feasible solution, and (ii) upon completion of the iterative algorithm, if (SP) admits no feasible solution.

In the former case, we can directly apply the Benders feasibility cuts (Birge and Louveaux 2011). Since our iterative algorithm solves the linear programming relaxation of (SP). Let $\boldsymbol{w}$ denote a feasible (MP) solution. The feasibility cuts are formulated as follows, for each scenario $s\in\mathscr{S}$:

$$0\geq \boldsymbol{\Lambda}_s\boldsymbol{b}(\boldsymbol{w},\boldsymbol{\phi_s}), \tag{B.8}$$

where $\boldsymbol{\Lambda}_s$ denotes an extreme ray for solutions of $\left(\text{D}-\overline{\text{SP}}\right)$ in scenario $s$. When $\left(\overline{\text{SP}}\right)$ admits no feasible solution, its dual $\left(\text{D}-\overline{\text{SP}}\right)$ has extreme rays $\boldsymbol{\Lambda}_s$—because $\boldsymbol{w}$ does not affect the feasibility region of $\left(\text{D}-\overline{\text{SP}}\right)$ and the feasibility region is nonempty (see Appendix B.1). This implies that $(\lambda\boldsymbol{\Lambda}_s)\boldsymbol{b}(\boldsymbol{w},\boldsymbol{\phi_s})\rightarrow+\infty$, because $(\lambda\boldsymbol{\Lambda}_s)$ is also a feasible dual solution for scalar $\lambda>0$. Equation (B.8) avoids this unbounded case, so $(\lambda\boldsymbol{\Lambda}_s)\boldsymbol{b}(\boldsymbol{w},\boldsymbol{\phi_s})\rightarrow-\infty$. This ensures the feasibility of $\left(\overline{\text{SP}}\right)$ with first-stage solution $\boldsymbol{w}$.

In the latter case, Proposition 2 provides a valid integer feasibility cut (Equation (7)). We prove it next.

**Proof of Proposition 2.** We need to show that Equation (7) is violated for solution $\widehat{\boldsymbol{w}}$ but not for any other feasible (MP) solution $\boldsymbol{w}$.

First, note that the vector $\boldsymbol{w}^{\text{dep}}$ is sufficient to determine a unique (MP) solution $\boldsymbol{w}$ due to Constraint (1d), and thus we omit the vector $\boldsymbol{w}^{\text{arr}}$ in the feasibility cut.

Let $\widehat{\boldsymbol{w}}$ denote a feasible (MP) solution such that the corresponding (SP) is infeasible in scenario $s\in\mathscr{S}$. Clearly, $\sum_{i\in\mathscr{F}}\sum_{t\in\tilde{\mathscr{T}}_i^{\text{dep}}}(\widehat{w}_{it}^{\text{dep}}-w_{it}^{\text{dep}})^2\geq 1$ is a valid feasibility cut, in that it is violated for solution $\widehat{\boldsymbol{w}}$ but not for any other solution $\boldsymbol{w}$. We deduct our integer feasibility cut from this expression, as follows:

$$\begin{aligned}
1 &\leq \sum_{i\in\mathscr{F}}\sum_{t\in\tilde{\mathscr{T}}_i^{\text{dep}}}(\widehat{w}_{it}^{\text{dep}}-w_{it}^{\text{dep}})^2,\\
&= \sum_{i\in\mathscr{F}}\sum_{t\in\tilde{\mathscr{T}}_i^{\text{dep}}}((\widehat{w}_{it}^{\text{dep}})^2-2\widehat{w}_{it}^{\text{dep}}w_{it}^{\text{dep}}+(w_{it}^{\text{dep}})^2),\\
&= \sum_{i\in\mathscr{F}}\sum_{t\in\tilde{\mathscr{T}}_i^{\text{dep}}}(\widehat{w}_{it}^{\text{dep}}-2\widehat{w}_{it}^{\text{dep}}w_{it}^{\text{dep}}+w_{it}^{\text{dep}}) \quad \text{because } (w_{it}^{\text{dep}})^2=w_{it}^{\text{dep}} \text{ and } (\widehat{w}_{it}^{\text{dep}})^2=\widehat{w}_{it}^{\text{dep}},\\
&= \sum_{i\in\mathscr{F}}\sum_{t\in\tilde{\mathscr{T}}_i^{\text{dep}}}w_{it}^{\text{dep}}-\sum_{i\in\mathscr{F}}\sum_{t\in\tilde{\mathscr{T}}_i^{\text{dep}}}\widehat{w}_{it}^{\text{dep}}w_{it}^{\text{dep}}+\sum_{i\in\mathscr{F}}\sum_{t\in\tilde{\mathscr{T}}_i^{\text{dep}}}(1-\widehat{w}_{it}^{\text{dep}})w_{it}^{\text{dep}}.
\end{aligned} \tag{B.9}$$

This completes the proof. □

### B.6. Additional Details on Local Branching and Pareto-optimality Cuts

We provide more details about our local branching and Pareto-optimality cuts adopted in Section 4.5.

**Local Branching.** Developed by Fischetti and Lodi (2003) to speed up IP solution algorithms, local branching divides the IP feasible region into small sub-regions, and derives a good feasible solution in each sub-region. Rei et al. (2009) apply this idea to Benders decomposition by solving the master problem via local branching, and using several first-stage solutions to generate several optimality and feasibility cuts at each iteration. This results in faster convergence of the decomposition algorithm.

Let $\Delta(\boldsymbol{w}^{(A)}, \boldsymbol{w}^{(B)}) = \sum_{i \in \mathscr{F}} \sum_{t \in \tilde{\mathscr{T}}^{\text{dep}}} |w_{it}^{\text{dep}(A)} - w_{it}^{\text{dep}(B)}|$ be the Hamming distance function between two feasible (MP) solutions $\boldsymbol{w}^{(A)}$ and $\boldsymbol{w}^{(B)}$. Let $\boldsymbol{w}^{(1)}$ be a feasible (MP) solution. We divide the feasible region of (MP) into two sub-regions: a "left" sub-region and a "right" sub-region with branching constraints $\Delta(\boldsymbol{w}, \boldsymbol{w}^{(1)}) \leq e$ and $\Delta(\boldsymbol{w}, \boldsymbol{w}^{(1)}) \geq e+1$, respectively, where $e > 0$. For $e$ sufficiently small, we can effectively solve the "left" sub-region. Let $\boldsymbol{w}^{(2)}$ denote the resulting solution—at least as good as $\boldsymbol{w}^{(1)}$. We further divide the "right" sub-region into two sub-sub-regions: a "left" sub-sub-region with branching constraints $\Delta(\boldsymbol{w}, \boldsymbol{w}^{(1)}) \geq e+1$ and $\Delta(\boldsymbol{w}, \boldsymbol{w}^{(2)}) \leq e$, and a "right" sub-sub-region with constraints $\Delta(\boldsymbol{w}, \boldsymbol{w}^{(1)}) \geq e+1$ and $\Delta(\boldsymbol{w}, \boldsymbol{w}^{(2)}) \geq e+1$. This procedure is repeated and, upon termination, the optimal solution is obtained from the solutions found in the "left" sub-regions and the remaining "right" sub-region. Ultimately, we extract the optimal solution $\boldsymbol{w}$ and the second-best one $\boldsymbol{w}'$, and pass them to $\left(\overline{\text{SP}}\right)$ to derive corresponding cuts. Note that we could extract more than two (MP) solutions at each iteration, but computational experiments suggest that it does not improve the algorithm's efficiency.

**Pareto-optimality Cut.** Given a feasible (MP) solution $\boldsymbol{w}$, Benders decomposition generates a cut that is "tight" in $\boldsymbol{w}$ but may not be "tight" for any other feasible (MP) solution. To address this issue, Magnanti and Wong (1981) derived non-dominated cuts (or "Pareto-optimal cuts") when the dual sub-problem admits several optimal solutions. Let $\boldsymbol{w}^{(o)}$ be a "core point" of the feasible region of (MP). The method selects the optimal solution of the dual sub-problem with $\boldsymbol{w}$ that yields the "tightest" cut at $\boldsymbol{w}^{(o)}$, thus guaranteeing that the cut is non-dominated in $\boldsymbol{w}$ and $\boldsymbol{w}^{(o)}$.

To implement this method, we need to solve the dual sub-problem $\left(\text{D}-\overline{\text{SP}}\right)$ (twice) but this is highly time-consuming. Indeed, facet-defining constraints (2b) and (2c) improve the computational efficiency of the primal (SP) and $\left(\overline{\text{SP}}\right)$ problems; however, these constraints are lost in the dual formulation $\left(\text{D}-\overline{\text{SP}}\right)$. To circumvent this challenge, we alternatively implement the perturbation scheme from Sherali and Lunday (2013). This approach involves solving $\left(\overline{\text{SP}}\right)$ once rather than solving $\left(\text{D}-\overline{\text{SP}}\right)$ twice. This is achieved by perturbing the right-hand side of the constraints of $\left(\overline{\text{SP}}\right)$ (Equation (3)) as follows:

$$\boldsymbol{A}_1\boldsymbol{x}_s + \boldsymbol{A}_2\boldsymbol{v}_s + \boldsymbol{E}\boldsymbol{y}_s = \boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \zeta \boldsymbol{b}(\boldsymbol{w}^{(o)}, \boldsymbol{\phi}_s), \tag{B.10}$$

where $\zeta$ is a small perturbation coefficient (set to $10^{-6}$), and $\boldsymbol{w}^{(o)}$ is a core point of (MP). The objective of $\left(\text{D}-\overline{\text{SP}}\right)$ is then $\boldsymbol{\Pi}_s\boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s) + \zeta\boldsymbol{\Pi}_s\boldsymbol{b}(\boldsymbol{w}^{(o)}, \boldsymbol{\phi}_s)$. Therefore, if $\zeta$ is chosen small enough, it will yield the optimal value of $\boldsymbol{\Pi}_s\boldsymbol{b}(\boldsymbol{w}, \boldsymbol{\phi}_s)$ first, and then maximize $\boldsymbol{\Pi}_s\boldsymbol{b}(\boldsymbol{w}^{(o)}, \boldsymbol{\phi}_s)$ as a secondary objective. In this study, we define $\boldsymbol{w}^{(o)}$ as the optimal solution of the LP relaxation of (IMSOAN). Note that the core point is usually defined as $\boldsymbol{w}^{(o)} = \mathbf{0.5}$. Based on initial computational experience, defining $\boldsymbol{w}^{(o)}$ by solving the LP relaxation of (IMSOAN) is slightly more efficient.

## Appendix C: Appendix to Scenario Generation (Section 5)

### C.1. Proof of Proposition 3

We assume that $d_{s'_1,s'_2} > 0$ for all $s'_1 \neq s'_2 \in \mathscr{S}'$, and consider an optimal solution $(\boldsymbol{y}^*, \boldsymbol{x}^*)$. To show statement (i), let us assume by contradiction that, for some $s \in \mathscr{S}$, we have $y^*_s = 1$ and $x^*_{s,s} = 0$. Per Equation (10d), there exists $s_0 \in \mathscr{S}'$ such that $x_{s,s_0} = 1$. Then we define another solution $(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{x}})$ such that $\tilde{y}_s = y^*_s$ for all $s \in \mathscr{S}'$, $\tilde{x}_{s,s} = 1$, $\tilde{x}_{s,s_0} = 0$, and $\tilde{x}_{s',s} = x^*_{s',s}$ for all $(s',s) \in \mathscr{S}' \times \mathscr{S} \setminus \{(s,s),(s,s_0)\}$. In other words, we shift the assignment of scenario $s$ from scenario $s_0$ to scenario $s$. This solution is clearly feasible, and we have $\sum_{s',s\in\mathscr{S}'} d_{s',s}\tilde{x}_{s',s} = \sum_{s',s\in\mathscr{S}'} d_{s',s}x^*_{s',s} - d_{s,s_0}$. This contradicts the optimality of $(\boldsymbol{y}^*, \boldsymbol{x}^*)$.

To show statement (ii), we suppose by contradiction that $\sum_{s\in\mathscr{S}'} y^*_s < S$. We define a subset $\mathscr{S}_0 \subset \mathscr{S}'$ such that $y^*_{s_0} = 0$ for all $s_0 \in \mathscr{S}_0$ and $|\mathscr{S}_0| = S - \sum_{s\in\mathscr{S}'} y^*_s$. Note that the subset $\mathscr{S}_0$ is not uniquely defined, but one can select any subset that satisfies this property. We define another solution $(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{x}})$ by adding the scenarios in $\mathscr{S}_0$ to the subset $\mathscr{S}$ and assigning these scenarios to themselves, as follows:

$$
\begin{aligned}
&\tilde{y}_s = 1 && \text{for all } s \in \mathscr{S}' \text{ such that } y^*_s = 1,\\
&\tilde{y}_{s_0} = 1 && \text{for all } s_0 \in \mathscr{S}_0,\\
&\tilde{y}_s = 0 && \text{for all } s \notin \mathscr{S}_0 \text{ such that } y^*_s = 0,\\
&\tilde{x}_{s_0,s_0} = 1 && \text{for all } s_0 \in \mathscr{S}_0,\\
&\tilde{x}_{s_1,s_2} = x^*_{s_1,s_2} && \text{for all } (s_1,s_2) \in (\mathscr{S}' \times \mathscr{S}') \setminus (\mathscr{S}_0 \times \mathscr{S}_0).
\end{aligned}
$$

This solution is clearly feasible, and we have $\sum_{s',s\in\mathscr{S}'} d_{s',s}\tilde{x}_{s',s} = \sum_{s',s\in\mathscr{S}'} d_{s',s}x^*_{s',s} - \sum_{s_0\in\mathscr{S}_0} d_{s_0,s_0}$. It contradicts the optimality of $(\boldsymbol{y}^*, \boldsymbol{x}^*)$. This completes the proof. $\square$