Extended Aircraft Arrival Management under Uncertainty: A Computational Study

Ahmed Khassiba, Fabian Bastin, Bernard Gendron, Sonia Cafieri, Marcel Mongeau

Extended Aircraft Arrival Management under Uncertainty: A computational study

Ahmed Khassiba*
ENAC, Université de Toulouse, FRANCE, & DIRO, Université de Montréal, CIRRELT, CANADA

Fabian Bastin†, Bernard Gendron‡
DIRO, Université de Montréal, CIRRELT, CANADA

Sonia Cafieri§, Marcel Mongeau¶
ENAC, Université de Toulouse, FRANCE

Arrival Manager operational horizon, in Europe, is foreseen to be extended up to 500 nautical miles around destination airports. In this context, arrivals need to be sequenced and scheduled a few hours before landing, when uncertainty is still significant. A computational study, based on a two-stage stochastic program, is presented and discussed to address the arrival sequencing and scheduling problem under uncertainty. This preliminary study focuses on a single Initial Approach Fix and a single runway. Different problem characteristics, optimization parameters as well as fast solution methods for real-time implementation are analyzed in order to evaluate the viability of our approach. Paris Charles-De-Gaulle airport is taken as a case study. A simulation-based validation experiment shows that our approach can decrease the number of expected conflicts near the terminal area by up to 70%. Moreover, in a high-density traffic situation, the total time-to-lose inside the terminal area can be decreased by more than 71%, while the expected landing rate can be increased by 7.7%, compared to the first-come first-served policy. This computational study demonstrates that sequencing and scheduling arrivals under uncertainty, a few hours before landing, can successfully diminish the need for holding stacks by relying more on upstream linear holding.

I. Introduction

Air traffic world-wide growth puts more and more pressure on major airports to better use their infrastructure in order to meet the required levels of safety and efficiency. Since the early 90's in the USA and Europe, air traffic controllers (ATCs) around major airports have been using decision support tools that compute "optimal" sequences and schedules of landings at the available runways. In Europe, such tools are called Arrival Managers (AMANs). AMAN

*PhD candidat, ENAC (Université de Toulouse) and DIRO (Université de Montréal), khassiba@iro.umontreal.ca.
†Associate Professor, DIRO, Université de Montréal, Montréal, Québec, H3T 1J4, CANADA, bastin@iro.umontreal.ca.
‡Full Professor, DIRO, Université de Montréal, Montréal, Québec, H3T 1J4, CANADA, gendron@iro.umontreal.ca.
§Professor, ENAC, 7 avenue Edouard-Belin BP 54005, 31055, Toulouse cedex 4, FRANCE, sonia@recherche.enac.fr.
¶Professor, ENAC, 7 avenue Edouard-Belin BP 54005, 31055, Toulouse cedex 4, FRANCE, mongeau@recherche.enac.fr.

typically captures inbound aircraft at a distance of 100-200 nautical miles (NM) around the destination airport (30-45 minutes before landing). In the last few years, extending AMAN operational horizon up to 500 NM (2 to 3 hours before landing) has been identified as one key measure to limit delays and to enhance punctuality and eco-efficiency [1]. In fact, starting the sequencing and scheduling process earlier will help aircraft fly more fuel-efficient trajectories and avoid congestion in the terminal area. The foreseen European tool is often referred to as Extended AMAN (E-AMAN). This new decision support tool is expected to diminish the need for *holding patterns* near the terminal area, a stressful ATC technique for both controllers and pilots in top of being extremely eco-inefficient. However, when the sequencing-and-scheduling horizons are extended, uncertainties about predicted times of arrival get large and cannot be overlooked. A possible technique to hedge against uncertainty is to re-optimize as soon as input data is updated. However, under large uncertainty, re-optimizing a deterministic model each time input data is updated, will deliver unstable solutions, making their implementation impractical. Instead of frequently re-optimizing deterministic models, uncertainties can be embedded within the optimization model in order to obtain more stable solutions. In this paper, we present a computational study based on a two-stage stochastic program addressing the problem of arrival sequencing and scheduling under uncertainty, two to three hours look-ahead time. Numerical tests on realistic instances from Paris Charles-De-Gaulle airport (CDG) are presented and discussed. In the following, the related literature is briefly reviewed before giving the paper outline.

Literature review

The problem of sequencing and scheduling arrivals on a given destination airport has been studied for several decades [2, 3]. Sequencing consists in finding an order among the considered aircraft, while scheduling is related to the timing of aircraft landings. Optimality criteria usually include maximizing runway throughput and/or minimizing aircraft delay. Operational constraints, mainly minimum separations called *final approach separations*, have to be satisfied between aircraft near the runway threshold. Variants of this problem may consider a single or multiple runways [4–6]. Also, when more operations (departures, runway crossings, etc) are included, the problem is often called the Aircraft Sequencing and Scheduling problem (ASSP). The case in which predicted operations times are known with certainty, called the *deterministic* case, has been thoroughly studied in the literature [4, 7, 8], while the case under uncertainty has less often been addressed. So far in the related literature, three main approaches to optimization under uncertainty were applied to ASSP: probabilistic [9, 10], stochastic [11–13] and robust [14–16] approaches. Pioneer studies such as [9, 10] mainly enriched deterministic models by probability constraints and/or by a probability objective-value function. Stochastic programming models, including two-stage and multi-stage models, were proposed in [11–13]. Finally, [14–16] proposed and studied several robust programming models for the runway scheduling problem. Apart from Kapolke et al. [16], all of the aforementioned studies concentrated on variants of ASSP involving a short planning horizon of around 45 minutes. Kapolke et al. [16] addressed the pre-tactical aircraft landing problem under uncertainty,

where aircraft are captured a few hours before landing.

Contribution and paper outline

In the context of the present paper, we seek to optimally sequence and schedule aircraft over the same Initial Approach Fix (IAF), two to three hours before landing. We propose a computational study based on a two-stage stochastic programming model. In the first stage, an aircraft sequence and a schedule at the IAF are found so as to maximize the expected runway throughput under uncertain arrival times at the IAF. In the second stage, uncertainty is assumed to be revealed and the landing times are computed so as to minimize a time-deviation impact cost in the terminal area. Focusing on realistic instances from CDG airport, we study the compromise between flexibility and punctuality when scheduling at the IAF as well as the effect of uncertainty amplitude. We examine fast solution methods from the literature and compare them to a different proposed approach under a limited computing-time budget. The aim of our computational study is to evaluate various model and optimization-method parameters as well as different resolution approaches, within the framework of two-stage stochastic programming, in order to design an efficient algorithm for real-time implementation. Also, since the first-come first-served (FCFS) policy is widely used in practice to schedule arrivals from the IAF to the runway threshold, we evaluate the benefit of our approach (sequencing and scheduling arrivals at the IAF with few hours look-ahead time) for the FCFS policy performance in the terminal area.

The remaining of this paper is organized as follows. First, the problem is described in Section II. The solution method is explained in Section III. The computational study is presented and discussed in Section IV. Finally, conclusions are drawn in Section V.

II. Problem statement

We consider a set of n aircraft planning to land at a given destination airport in two to three hours look-ahead time. Let $\mathcal{A} = \{1, \ldots, n\}$ be the set of their indices. We make the following two operational assumptions. Firstly, all aircraft will fly to the same IAF before entering the airport terminal area. Secondly, all aircraft will land on the same runway of the considered airport. We are given a *predicted time at the IAF* for each aircraft. We seek to sequence and schedule these aircraft to the IAF so as to maximize the expected landing rate as well as to minimize a time-deviation impact cost in the terminal area. To that aim, we first introduce the continuous decision variable x_i as the *target time at the IAF* for aircraft $i \in \mathcal{A}$. Target times at the IAF must satisfy two types of constraints detailed below: minimum-time separation at the IAF, and time-windows constraints.

Minimum-time separation at the IAF

We assume that all aircraft pass over the IAF at the same altitude (*i.e.* the same flight level) so that only longitudinal separation between aircraft matters. Aircraft successively passing over the IAF have to be separated by a distance-based

minimum separation. For modeling and optimization purposes, this minimum separation at the IAF is converted to time. Let us note $\underline{S}^I$ the time-based minimum separation at the IAF expressed in seconds. Assuming all aircraft speeds over the IAF are equal to 250 knots and a distance-based minimum separation at the IAF of 5 NM, $\underline{S}^I$ may then be set to 72 seconds [9].

Time windows at the IAF

Imposing a time window constraint at some point of an aircraft trajectory either reflects its physical limitations (mainly its lowest and highest eligible speeds) or acceptable deviations with respect to some reference time as may be expressed by airlines or in order to ensure some level of air traffic punctuality. In our context, a target time at the IAF for each aircraft has to be found within a predefined time window, noted TW^I, around the predicted arrival time at the IAF.

We assume that actual times at the IAF will randomly deviate from the target times following a normal distribution with mean μ and standard deviation σ. In order to define the aircraft sequence over the IAF, we then introduce binary decision variables δ_{ij} for each pair of aircraft (i, j) such that $i \neq j$:

$$\delta_{ij} = \begin{cases} 1 & \text{if aircraft } i \textbf{ directly precedes} \text{ aircraft } j \\ 0 & \text{otherwise} \end{cases}$$

From the ATC perspective, aircraft will ideally pass over the IAF in the same sequence that maximizes the landing rate so that ATCs in the terminal area will only have to "compress" the sequence, without shifting any aircraft position inside the terminal area. This may be achieved by enforcing the target sequence over the IAF to be an optimal landing sequence. An optimal landing sequence is one that has a minimal length in terms of final-approach separations. Let us note S_{ij} the minimum time-based final approach separation in seconds between a leading aircraft i and a following aircraft j. Final-approach separations are defined in terms of distance in NM according to the wake-turbulence categories (Heavy (H), Medium (M) and Light (L)) set by the International Civil Aviation Organization (ICAO). Final-approach separations applicable in CDG airport and converted into seconds are given in Table 1.

Table 1 Final-approach separations (seconds) according to ICAO wake-turbulence categories [17]

		Following aircraft		
		H	M	L
Leading aircraft	H	96	157	207
	M	60	69	123
	L	60	69	82

Given the binary decision variables introduced above, the landing sequence length may be computed using the

following expression:

$$\sum_{\substack{(i,j)\in\mathcal{A}\times\mathcal{A}\\ i\neq j}} \delta_{ij} S_{ij}$$

Besides anticipating the optimal landing sequence, making aircraft arrive as early as possible at the IAF is equivalent to minimizing their average flight time during the en-route and descent phases, prior to the IAF. Such an objective may help flights catch up with their upstream delay, if any, in an attempt to improve punctuality. For that reason, we may try to minimize the sum of target times at the IAF, $\sum_{i\in\mathcal{A}} x_i$. Overall, we seek to minimize the following objective function, where x denotes the vector whose ith component is x_i for $i \in \mathcal{A}$, δ stands for the matrix whose ij-entry is δ_{ij} for $(i, j) \in \mathcal{A} \times \mathcal{A}$ such that $i \neq j$ and λ is a user-defined weighting parameter:

$$f_1(x, \delta) = \sum_{\substack{(i,j)\in\mathcal{A}\times\mathcal{A}\\ i\neq j}} \delta_{ij} S_{ij} + \lambda \sum_{i\in\mathcal{A}} x_i \tag{1}$$

In order to account for the expected air traffic situation in the terminal area, we consider a hypothetical second-stage problem in which actual arrival times at the IAF are assumed to be known with certainty. The beginning of the second stage can be set to the entry time of the last considered aircraft to the en-route sector neighbouring the terminal area. Defined as such, the second-stage problem corresponds to a deterministic aircraft sequencing-and-scheduling problem with a short operational horizon. In this second-stage problem, we schedule aircraft to the runway threshold so as to minimize the total time-deviation impact cost during the approach phase in view of eliminating congestion in the terminal area and improving punctuality.

The landing sequence is enforced to be the same as the already-found target sequence over the IAF, although the actual sequence over the IAF may be different once uncertainty is revealed. This is due to the fact that deviations of actual times at the IAF with respect to the target times may change the actual sequence over the IAF with respect to the target sequence. Since the landing sequence is already found in the first stage, no sequencing variables are needed in the second stage. Hence, we introduce the continuous decision variable y_i as the *target landing time* of aircraft $i \in \mathcal{A}$. Similarly to the target arrival times at the IAF, target landing times should satisfy two types of constraints: minimum time-based final-approach separation and time-windows constraints for landing. We define U_i the *unconstrained landing time* of aircraft i corresponding to the landing time of aircraft i as if it were alone in the terminal area flying its preferred trajectory at its preferred speed. For each aircraft $i \in \mathcal{A}$, U_i is computed as the sum of its actual arrival time at the IAF and an *unconstrained flight time* from the IAF to the runway threshold. Remark that the unconstrained landing time, computed at the beginning of the second stage, can be seen as the latest up-to-date estimated landing time available at around 30-minute look-ahead time. This unconstrained landing time is defined so that aircraft i flies its preferred trajectory at its preferred speed in the terminal area until landing, thereby saving fuel, and landing with no extra delay

incurred during the approach phase. Hence, $|y_i - U_i|$ quantifies the (unweighted) impact cost of time deviation from the unconstrained landing time of aircraft i during the approach phase. In this paper, we assume that deviating from the unconstrained landing time in either directions is equally undesirable. For a study with weighted time-deviation impact costs during the approach phase, see [18]. In the context of AMAN, a time deviation from the unconstrained landing time of a given aircraft can be seen as the *time-to-gain* or the *time-to-lose* as computed by AMAN, and then displayed to the approach controller. The controller has, then, to apply adequate control actions in order to satisfy this time deviation. Remark that if all inbound flights are able to follow their preferred trajectories at their preferred speed in the terminal area, then such a traffic situation should generate a low workload for both terminal-area controllers and pilots. Such an ideal situation is necessarily, but not sufficiently, described by the equation $\sum_{i \in \mathcal{A}} |y_i - U_i| = 0$. Indeed, an aircraft may succeed to land at its unconstrained time (*i.e.*, $y_i = U_i$), after stretching its path, speeding up, and then slowing down, generating thereby a significant workload for both pilots and controllers.

We define a *second-stage scenario* s as a possible realization of actual arrival times at the IAF of all aircraft in $\mathcal{A}$. For example, consider a set of 3 aircraft with IAF target times 6:00:00 AM, 6:01:30 AM and 6:03:00 AM for aircraft 1, 2 and 3 respectively. One possible realization of their IAF actual times, *i.e.* one possible scenario, is 6:00:32 AM, 6:01:10 AM and 6:03:55 AM respectively. In this scenario, the first and the third aircraft arrive at the IAF later than their target times (by 32 and 55 seconds respectively), while the second aircraft arrives 20 seconds earlier than its target time. Assuming 11 minutes of unconstrained flight time from the IAF to the runway threshold for any of the three aircraft, their unconstrained landing times in this scenario are 6:11:32 AM, 6:12:10 AM and 6:14:55 AM respectively. Remark that unconstrained landing times and target landing times depend on the scenario s. Hence, unconstrained landing time and target landing time of aircraft i in scenario s will be noted U_i^s and y_i^s respectively. Therefore, given a scenario s, the second-stage problem reads:

$$
\begin{aligned}
Q(x, \delta, s) := \quad & \min_{y^s} \quad \sum_{i \in \mathcal{A}} \left| y_i^s - U_i^s \right| \\
& \textit{subject to} \quad \text{final-approach separation constraints} \\
& \qquad\qquad\quad \text{and time-windows constraints for landing}
\end{aligned}
\tag{2}
$$

Note that, in our second-stage problem, we do not adjust the first-stage solution but we make new decisions (target landing times) based on the revealed uncertainty.

Since we need to account for all possible scenarios in the second-stage, the objective-function of the entire two-stage stochastic programming model reads:

$$
\min_{x, \delta} \quad f_1(x, \delta) + \mathbb{E}\left[Q(x, \delta, .)\right] \tag{3}
$$

where $\mathbb{E}[.]$ is the expectation operator. In the sequel, we shall need the following notation. Let $v^\star$ be the optimal value of the two-stage stochastic problem. The output of the two-stage model is then an optimal sequence (described by δ_{ij} for all $(i, j) \in \mathcal{A} \times \mathcal{A}$ such that $i \neq j$) and optimal target times at the IAF (x_i for all $i \in \mathcal{A}$) that maximize the expected landing rate and minimize the expected second-stage cost function.

We remark that the above model can be extended to the case of multiple IAFs at the expense of adding extra binary variables and associated constraints for the newly considered IAFs. A similar extension can be envisaged for the case of multiple runways.

The solution method proposed to address the two-stage stochastic program is explained in the next section.

III. Solution method

Two-stage stochastic programming models are usually intractable if all possible scenarios are taken into account. Nevertheless, the Sample Average Approximation (SAA) method [19] offers a framework to approximate the solutions of such problems, through solving an approximate problem, called the SAA problem, instead of the original problem. In the approximate problem, the expectation term in the objective function is estimated through a sample average computed over a finite set of scenarios. Given a set $\mathcal{S}$ (called "*training set*") of $n_\mathcal{S}$ equiprobable scenarios, the objective function of the SAA problem reads:

$$\min_{x,\delta} \quad f_1(x, \delta) + \frac{1}{n_\mathcal{S}} \sum_{s \in \mathcal{S}} Q(x, \delta, s) \tag{4}$$

Let $\hat{v}(\mathcal{S})$ be the optimal value of the SAA problem. Since $\hat{v}(\mathcal{S})$ depends on the scenarios set $\mathcal{S}$, $\hat{v}(\mathcal{S})$ is, itself, a random variable. The SAA method [19] guarantees that for any given $n_\mathcal{S}$, $\mathbb{E}[\hat{v}(\mathcal{S})] \leq v^\star$, *i.e.* the optimal value of the SAA problem is negatively biased. Moreover, under mild conditions, as $n_\mathcal{S} \to \infty$, $\hat{v}(\mathcal{S})$ converges towards $v^\star$ with probability one. However, in practice, the required computing time grows rapidly with $n_\mathcal{S}$. One difficulty with the SAA method is to decide whether a given number of scenarios, $n_\mathcal{S}$, is large enough to correctly approximate the original problem, *i.e.* to ensure that the solution obtained is a satisfying approximation of an optimal solution of the original problem. $\hat{v}(\mathcal{S})$ is not necessarily a good-quality indicator with respect to the original problem due to the SAA bias and variance of the SAA optimal value. Hence, a post-optimization validation step is needed to evaluate the quality of any SAA solution. In our study, we rely on the so-called *out-of-sample validation*, that consists in re-evaluating an SAA solution, $\left(\hat{x}, \hat{\delta}\right)$, with a *validation set* containing much more scenarios than the training set used to find $\left(\hat{x}, \hat{\delta}\right)$. The validation set is believed to represent the complete set of all possible scenarios. An out-of-sample validation provides a new quality indicator for the considered solution, called the *validation score*. When the gap between the SAA optimal value $\hat{v}(\mathcal{S})$ and the validation score of $\left(\hat{x}, \hat{\delta}\right)$ is small, the SAA problem can be considered as stable and the training set used may be considered as large enough. Accordingly, in our exploratory computational study detailed in Section IV, we investigate solving a sequence of SAA problems involving increasing numbers of scenarios ($n_\mathcal{S}$) under a limited,

although large, solving time. As we mentioned earlier, SAA solutions depend on the random set $\mathcal{S}$ of scenarios used for optimization. Therefore, to illustrate better the average behavior of SAA problems with a given number of scenarios $n_{\mathcal{S}}$, we build $n_{\mathcal{R}}$ replicated SAA problems with $n_{\mathcal{S}}$ scenarios each. Let $\bar{v}\left(n_{\mathcal{S}}, n_{\mathcal{R}}\right)$ be the average optimal value obtained over $n_{\mathcal{R}}$ such replications. It can be expressed as follows, where $\mathcal{S}_r$ is a scenario set such that $|\mathcal{S}_r| = n_{\mathcal{S}}$:

$$\bar{v}\left(n_{\mathcal{S}}, n_{\mathcal{R}}\right) = \frac{1}{n_{\mathcal{R}}} \sum_{r=1}^{n_{\mathcal{R}}} \hat{v}\left(\mathcal{S}_r\right) \tag{5}$$

Although a well-applied SAA method can guarantee good-quality solutions for the original problem, computing times are very often inappropriate for real-time implementation. Consequently, fast solution methods based on SAA were proposed in the literature related to aircraft scheduling using stochastic programming [11, 13].

Real-time solution methods

For real-time implementation, mainly two approaches can be built on the SAA method: replication-based and scenario-based approaches.

Replication-based approach

Since solving a single SAA problem with a large scenario set may be time-consuming, [11, 13] opt for solving several replications of an SAA problem with a limited number of scenarios. Upon optimization, they are left with a pool of near-optimal solutions to the original problem (as many solutions as replications). Distinct solutions are kept in a pool from which only one solution need to be selected at the end. On one hand, Bosson and Sun [13] simply select the solution with the minimum objective function value. Hence, no post-optimization validation is performed. On the other hand, Sölveling et al. [11] re-evaluate all distinct solutions from the pool using as a validation set, the complete set of scenarios of their original problem. Then, they select from the pool the solution with the best validation score. Let us note that increasing the number of replications may enlarge the solutions' pool. However, it does not guarantee better solutions.

Scenario-based approach

Unlike replication-based approach, in a scenario-based approach, only one SAA problem with a large enough number of scenarios is solved. Hence, no replications are made. The quality of the obtained solution can be estimated through out-of-sample validation. Let us note that increasing the number of scenarios often leads to better quality solutions.

IV. Computational study

We rely on a two-stage stochastic program implemented in Julia programming language [20] and on CPLEX 12.6.3 solver. Results are obtained on a Linux platform with 8 x 2.66 GHz Xeon processors and 32 GB of RAM. As a

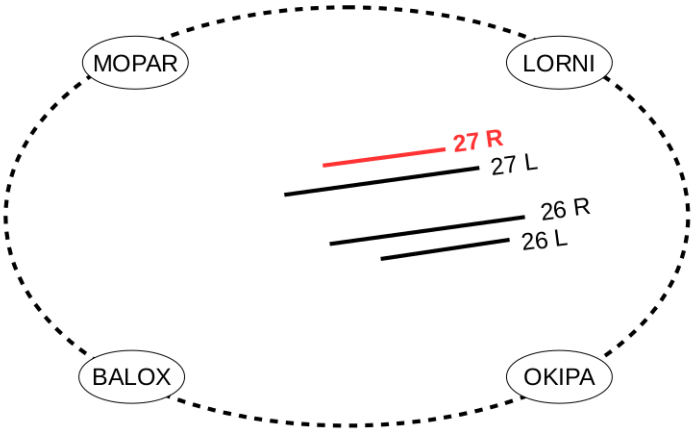
study case, we select the arrivals that planned to enter the terminal area around CDG airport on May 15th, 2015 from 6:00 AM to 6:30 AM and that landed on the north runway (27R). Two realistic instances, involving n =10 and 14 aircraft respectively, are extracted. To conduct our computational study, we consider different problem characteristics and optimization parameters. First, we try to find the best combination of these characteristics and parameters by intensive empirical experiments on the first instance (n =10 aircraft). Once the best combination is identified, we compare two real-time solution methods under a fixed solving-time budget. Using our best setting, we apply it to the second instance (n =14 aircraft). Finally, we evaluate the benefit of our approach (sequencing and scheduling arrivals at the IAF under uncertainty) for the expected performance of a FCFS policy in the terminal area. To that end, different performance indicators are computed through simulation-based experiments.

The two instances and the different problem characteristics are described in Subsection IV.A. The optimization parameters along with their tested values are presented in Subsection IV.B. The main results for the first instance are reported and discussed in Subsection IV.C. To select an effective real-time solution method, we compare the performance of the scenario-based and the replication-based approaches on the first instance in Subsection IV.D. Results for the second instance (n =14 aircraft) are presented in Subsection IV.E. Finally, the benefit of our approach on the FCFS policy in the terminal area is discussed in Subsection IV.F.

IV.A Instances and problem characteristics

The terminal area around CDG has four IAF named: MOPAR, LORNI, OKIPA and BALOX. CDG has four runways: two for landings (27R and 26L) and two for departures (27L and 26R). A simplified scheme of CDG runways with the surrounding IAFs is displayed in Figure 1.

Fig. 1 CDG runways scheme with the four surrounding IAFs (not to scale)



On May 15th 2015, looking only at aircraft that finally landed on CDG runway 27R, 10 of these aircraft were planned

to enter the terminal area between 6:00 AM and 6:20 AM, while 14 were planned to do so between 6:10 AM and 6:30 AM. All of these aircraft entered the terminal area from three different IAFs (MOPAR, LORNI and OKIPA). For the sake of simplification, we merge all these arrivals as if they were planned to pass over a single IAF. We are then left with two realistic instances satisfying our operational context (a single IAF and a single runway). Details of these instances are summarized in Table 2.

Table 2 The two considered instances

		instance 1	instance 2
Planned time span at the IAF		6:00 – 6:20	6:10 – 6:30
Number of aircraft per original IAF	MOPAR	7	8
	LORNI	2	5
	OKIPA	1	1
Total number of aircraft		10	14
Wake-turbulence category mix		H : 70% M : 30%	H : 50% M : 50%

Next, two problem characteristics are explored: the width of time windows at the IAF and the uncertainty amplitude.

Time windows at the IAF

The width of the time windows at the IAF may reflect different desired levels of flexibility and punctuality. Wide time windows offer more flexibility to optimize the sequence and the schedule at the IAF, whereas narrow time windows yield more punctuality. From an operational perspective, if an aircraft with a ground speed of 450 kts increases its speed by 3%, it will only save 1 minute with respect to its planned time over a distance of 300 NM. Therefore, in our study, we limit the acceptable time advance with respect to the IAF planned times to 1 minute. However, different levels of acceptable delays can be defined. For example, in [7], delays are allowed to reach one hour, while time advances are limited to 1 minute. Following the XMAN concept (a first operational step towards E-AMAN implementation), 5 minutes of delay can be achieved in the en-route and descent phases using only speed reduction over 300 NM. This defines a narrow IAF time window. For wide IAF time windows, we assume 10 minutes of feasible additional delay using path stretching. To summarize, in our study, the tested time windows are: [-1 min, +5 min] (narrow) and [-1 min, +15 min] (wide).

Uncertainty

Following the literature ([9, 21]), deviations of the actual times with respect to the target times at the IAF are assumed to follow a normal distribution with mean 0 and some standard deviation σ. To assess the impact of uncertainty, we test two values for σ: 30 seconds (small) and 60 seconds (large).

Other problem characteristics relevant to the second stage (time windows for landing and unconstrained flight time from the IAF to the runway threshold) are kept constant. For every aircraft, we opt for [-1 min, +19 min] time windows for landing, and 11 minutes for the unconstrained flight time from the IAF to the runway threshold (regardless of the aircraft type). This time window is constructed based on the minimum flight time observed in our data (10 minutes) and a maximum flight time of 30 minutes, where the extra 20 minutes can be spent in a holding stack for example. Assuming one minute as the longest time advance within the terminal area, we set the unconstrained flight time to 11 minutes.

IV.B Optimization parameters

Two main optimization parameters are studied: the weighting parameter in the first-stage objective function, λ, and the number of second-stage scenarios, $n_{\mathcal{S}}$.

First-stage objective-function weight

In addition to minimizing the landing sequence length, different weights λ in the first-stage objective function are tested in order to evaluate the effect of minimizing the sum of target times at the IAF. Two weighting parameters values are defined ($\lambda = 0$, and $\lambda = 0.01$). With $\lambda = 0$, only the landing sequence length is minimized. With $\lambda = 0.01$, a compromise is considered between the landing sequence length and the sum of target times at the IAF, such that both quantities have comparable amplitudes.

Number of second-stage scenarios

Since we follow an exploratory approach that successively increases the number of scenarios, we solve SAA problems with $n_{\mathcal{S}} = 10$, 50, 100, 200, 500 and 1000.

IV.C Results for instance 1

Instance 1 is solved with all the 48 possible combinations (6 different numbers of scenarios, 2 IAF time-window widths, 2 uncertainty amplitudes, and 2 values for the weight λ). For each setting combination, $n_{\mathcal{R}} = 10$ replications are performed. Far each replication, the time limit in CPLEX solver is set to one hour. Although this time limit is inappropriate for real-time implementation, it was selected for exploration and study purposes as explained in Section III.

The main results for instance 1 are shown on Tables 3 to 6. "CPU" stands for CPLEX solving time expressed in seconds, averaged on all the replications. When the time limit is reached for all replications, "Tilim" is indicated instead of the exact time value. "Status" tells whether CPLEX proved the optimality of the feasible solutions found. Note that feasible solutions are found for all replications and under all settings. When solutions for all replications are proved optimal by CPLEX then "Opt." is reported, while "r Opt." or "Feas." are reported if only r ($1 \leq r < n_{\mathcal{R}}$) or no solutions are proved optimal, respectively. "Gap" stands for the average (over all replications) percentage error of the

Table 3 Results for instance 1 with narrow IAF time windows and $\lambda = 0$

	TW^I	Narrow	
n_S	σ	Small	Large
10	CPU (s)	0.3	0.5
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	934.8 ± 8.0	1028.3 ± 19.3
	Validation	950.8	1060.9
50	CPU (s)	1.3	2.2
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	938.9 ± 3.1	1034.9 ± 6.6
	Validation	944.2	1046.4
100	CPU (s)	3.2	6.4
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	940.4 ± 2.2	1040.3 ± 4.8
	Validation	943.8	1045.6
200	CPU (s)	9.6	18.5
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	940.3 ± 2.0	1039.6 ± 4.8
	Validation	942.6	1045.1
500	CPU (s)	75.1	144.3
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	941.6 ± 0.9	1042.8 ± 1.9
	Validation	942.0	1044.1
1000	CPU (s)	358.6	812.0
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	941.5 ± 0.8	1042.8 ± 1.8
	Validation	941.9	1043.7

best bound with respect to the value of the best feasible solutions returned by CPLEX. Here, $\bar{v}$ stands for the average objective value over the replications solved to optimality, noted $\bar{v}\,(n_S, n_R)$ in Section III. For each computed value of $\bar{v}\,(n_S, n_R)$, a 95%-confidence interval is computed, thanks to the central limit theorem, its radius is noted "$I_{95\%}$". In our context, a 95%-confidence interval indicates that we are 95% sure that the true value of $\mathbb{E}\left[\hat{v}\,(S)\right]$ lies within this interval. "Validation" stands for the out-of-sample validation score, introduced in Section III, averaged on all the replications. The validation set is made of 10,000 scenarios. Figures 2 to 5 plot $\bar{v}$, "Validation" and CPU as functions of n_S from Tables 3 and 5. Box-plots around $\bar{v}$ and validation scores are also shown in order to give some insight on the distribution of replication-specific values.

Effect of the number of scenarios, n_S

As expected, the solving time increases rapidly with n_S. In 18 out of 480 runs, CPLEX is not able to prove optimality within 1 hour. These runs correspond to tests on instance 1 with wide IAF time windows, under large uncertainty and $n_S = 1000$ scenarios. Nevertheless, we observe that, in 6 out of 8 cases, we can solve replications with up to $n_S = 500$

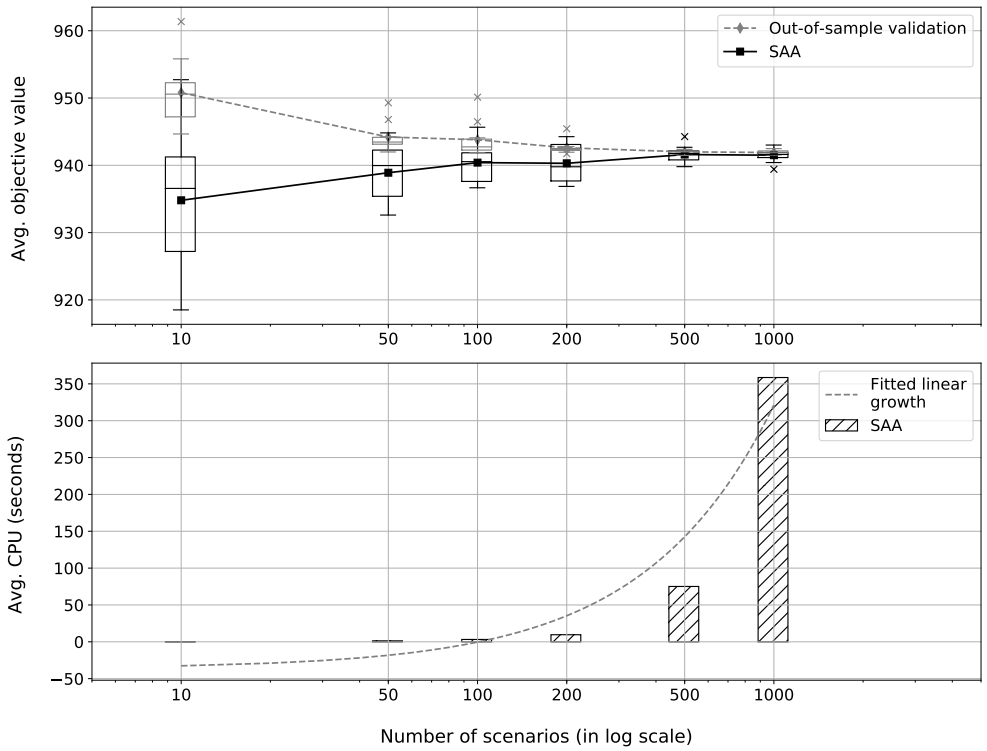


Fig. 2 Average objective-function values and validation scores (upper figure) and average CPU times (lower figure) for instance 1 with narrow IAF time windows, $\lambda = 0$ and small uncertainty.

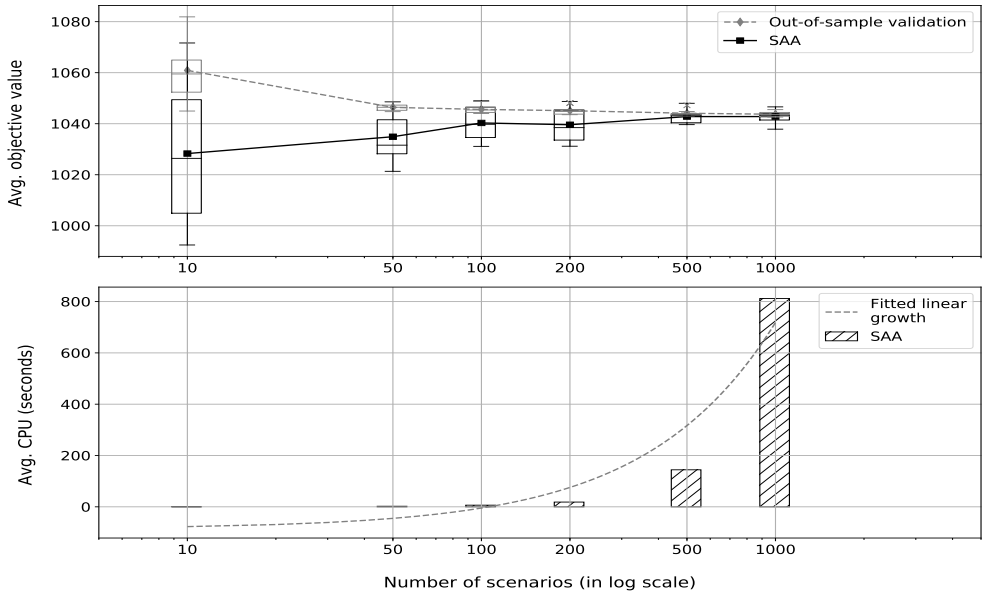


Fig. 3 Average objective-function values and validation scores (upper figure) and average CPU times (lower figure) for instance 1 with narrow IAF time windows, $\lambda = 0$ and large uncertainty.

scenarios in less than 3 minutes, on average. In the two remaining cases (corresponding to instance 1 with wide IAF time windows, under large uncertainty), we can solve replications with up to $n_{\mathcal{S}} = 200$ scenarios in less than 4 minutes.

Table 4 Results for instance 1 with narrow IAF time windows and $\lambda = 0.01$

	TW^I	Narrow	
n_S	σ	Small	Large
10	CPU (s)	0.3	0.5
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1730.9 ± 8.0	1823.5 ± 18.5
	Validation	1745.9	1852.7
50	CPU (s)	1.5	2.5
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1733.5 ± 3.2	1829.6 ± 6.8
	Validation	1737.9	1841.1
100	CPU (s)	3.8	6.8
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1735.2 ± 2.2	1835.2 ± 4.8
	Validation	1737.7	1840.5
200	CPU (s)	11.1	20.7
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1734.8 ± 2.0	1834.4 ± 4.8
	Validation	1737.1	1839.9
500	CPU (s)	80.6	159.3
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1736.2 ± 0.9	1837.5 ± 1.9
	Validation	1736.5	1838.6
1000	CPU (s)	372.3	837.4
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1736.0 ± 0.7	1837.6 ± 1.8
	Validation	1736.4	1838.3

We observe that as the number of scenarios increases, average objective-function values, $\bar{v}$, clearly increase while the spread of objective-function values around the average decreases. These two facts correctly illustrate the behavior of SAA problems' objective-function values when increasing the number of scenarios. On the other hand, as expected, average validation scores decrease with the number of scenarios. This reveals that increasing the number of scenarios leads to better-quality solutions. Remark that, for any given number of scenarios n_S, the average validation score is greater than the average objective-function value. This demonstrates that the value of an optimal solution obtained with a limited number of scenarios is often underestimated.

For any given test setting, we do not observe any change of the landing sequence length when increasing the number of scenarios, while slight modifications of IAF target times occur. However, a significant increase is observed in terms of second-stage cost (as the number of scenarios increases). We conclude that, under our test settings, increasing the number of scenarios helps estimating more accurately the second-stage cost, and eventually adjusting IAF target times.

Table 5 Results for instance 1 with wide IAF time windows and $\lambda = 0$

	TW^I	Wide	
n_S	σ	Small	Large
10	CPU (s)	0.3	1.9
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	826.3 ± 0.7	876.8 ± 12.4
	Validation	840.2	929.7
50	CPU (s)	3.1	20.8
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	829.9 ± 1.1	894.2 ± 5.0
	Validation	834.3	907.8
100	CPU (s)	7.5	52.5
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	830.8 ± 0.8	897.7 ± 2.9
	Validation	833.7	906.9
200	CPU (s)	23.1	182.6
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	831.6 ± 0.6	899.4 ± 2.9
	Validation	833.4	906.3
500	CPU (s)	122.3	1023.2
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	832.3 ± 0.2	902.3 ± 1.4
	Validation	833.1	904.9
1000	CPU (s)	472.7	3557.9
	Status (Gap)	Opt. (0.0%)	2 Opt. (2.9%)
	$\bar{v} \pm I_{95\%}$	832.3 ± 0.1	902.2 ± 0.8
	Validation	833.0	904.7

Effect of time-window width at the IAF

For both values of the weighting parameter λ, it is clear that the problem with narrow time windows is easier to solve to optimality than with wide time windows. With relatively wide time windows, too much flexibility is left to the solver to find an optimal solution. More precisely, the maximum number of positions to which aircraft can be shifted grows with the width of the time windows. The additional sequences offered by wide time windows may achieve better values of the objective function, as it is the case when comparing Tables 3 and 5 (both with $\lambda = 0$) for example. In fact, the minimum sequence length (not shown in the tables) found with narrow time windows (for any values for the remaining test settings) is 887 seconds, while with wide time windows (for any values for the remaining test settings), optimal sequences have a length of 826 seconds. This can be explained by looking closer at the first three aircraft in instance 1, where the first and the third aircraft belong to the heavy turbulence category, while the second aircraft is a medium-turbulence jet. Due to narrow time windows, the first two aircraft cannot be shifted, which results in the partial sequence "H-M-H" whose length is 217 seconds. With wide time windows, the medium-turbulence-category aircraft can be shifted to the first position so that the partial sequence becomes "M-H-H", whose length is only 156 seconds.

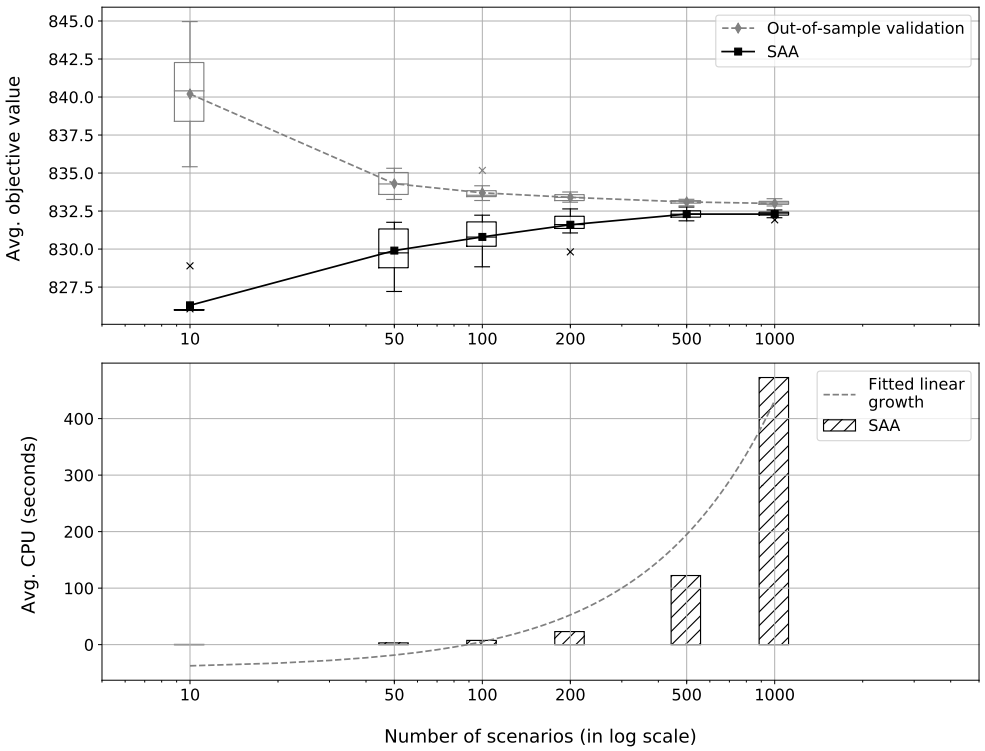


Fig. 4 Average objective-function values and validation scores (upper figure) and average CPU times (lower figure) for instance 1 with wide IAF time windows, $\lambda = 0$, small uncertainty.

Better values of the objective function come at the expense of relatively longer solving times, especially under large uncertainty. For example, while with narrow time windows and under large uncertainty SAA problems with $n_S = 500$ scenarios can be solved in less than 3 minutes on average, more than 17 minutes on average are needed to solve the same problems with wide time windows. On the other hand, in this example, using wide time windows improves the average objective-function value by 13%, and more precisely the expected landing sequence length is shortened by 61 seconds, compared to the case with narrow time windows.

In terms of second-stage cost, for $n_S = 1000$ scenarios, we remark that, with wide IAF time windows, the average second-stage cost is decreased when compared to the test with narrow IAF time windows. In fact, wide IAF time windows allow more spaced IAF target times, which help absorbing the effect of uncertainty when revealed.

Finally, in our context, narrow time windows may be preferred to wide ones because they are likely to boost flight on-time performance and to reduce fuel consumption.

Effect of uncertainty amplitude

Throughout all the results, as uncertainty gets larger, more time is needed to solve the problem. Concerning runs with narrow time windows and different numbers of scenarios n_S, the average CPU time may increase up to 2 times when the uncertainty standard deviation is doubled. The increase factor is more important when we switch from narrow

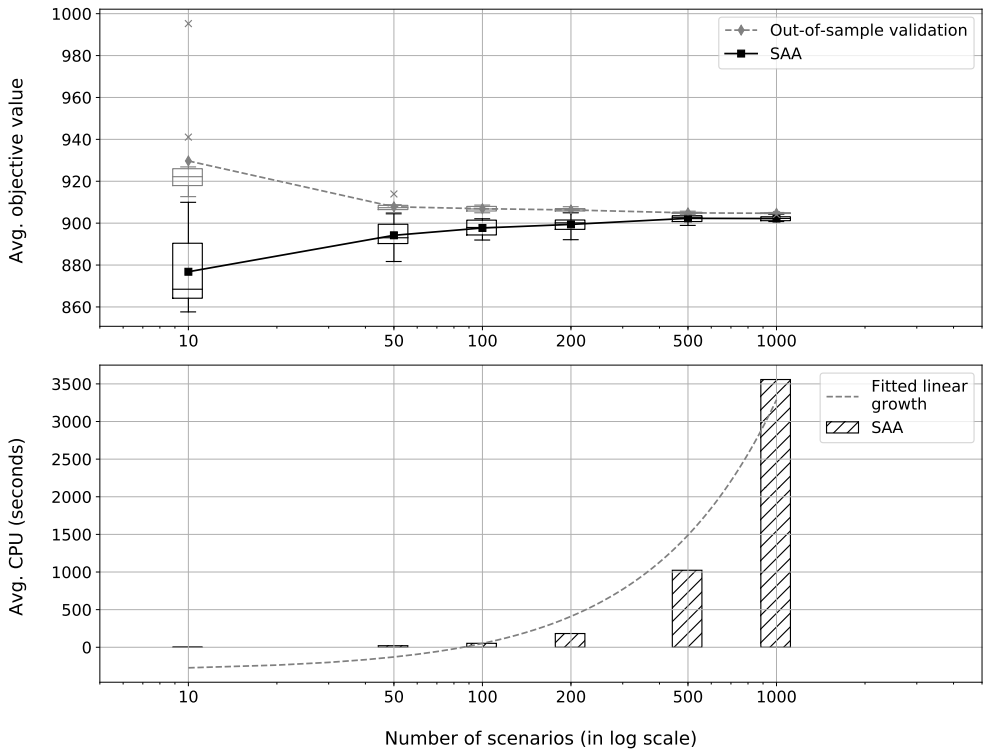


Fig. 5 Average objective-function values and validation scores (upper figure) and average CPU times (lower figure) for instance 1 with wide IAF time windows, $\lambda = 0$, large uncertainty.

to wide time windows. For example, with wide time windows and $n_{\mathcal{S}} = 500$, regardless of the value of λ, average CPU times increase by more than 8 times from small to large uncertainty.

Also, we observe that the gap between $\bar{v}$ and the validation score, called "validation" gap, decreases less rapidly under large uncertainty than under small uncertainty, as the number of scenarios increases. Consequently, we may conclude that as the uncertainty increases a larger number of scenarios is needed to correctly represent the original problem. Since computation times increase as uncertainty and the number of scenarios increase, real-time implementation of a stochastic programming approach based on scenario sampling may be very challenging.

Tested values of uncertainty standard deviation reveal no effect of uncertainty amplitude on the landing sequence length. However, larger uncertainty amplitudes are worthwhile to be tested. On the other hand, second-stage cost significantly increases as uncertainty increases. Figure 6 shows the effect of uncertainty amplitude and IAF time windows on the average second-stage cost, with $\lambda = 0$, $n_{\mathcal{S}} = 1000$ scenarios, and both narrow and large IAF time windows.

Effect of minimizing the sum of target times at the IAF in the first stage

We observe that, when the weighted sum of target times at the IAF is also minimized in the first stage, average solving times slightly increase. The increase factor is greater with wide time windows than with narrow ones. For

Table 6 Results for instance 1 with wide IAF time windows and $\lambda = 0.01$

	TW^I	Wide	
n_S	σ	Small	Large
10	CPU (s)	0.7	2.6
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1637.4 ± 2.4	1691.6 ± 12.1
	Validation	1657.6	1743.4
50	CPU (s)	5.0	23.9
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1643.9 ± 1.3	1709.1 ± 5.2
	Validation	1648.9	1723.2
100	CPU (s)	13.8	65.3
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1645.0 ± 1.1	1712.5 ± 2.8
	Validation	1647.9	1721.9
200	CPU (s)	33.9	217.7
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1646.0 ± 0.9	1714.3 ± 2.9
	Validation	1647.8	1720.9
500	CPU (s)	164.9	1351.4
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1646.5 ± 0.3	1717.1 ± 1.5
	Validation	1647.3	1719.8
1000	CPU (s)	661.6	Tilim
	Status (Gap)	Opt. (0.0%)	Feas. (2.8%)
	$\bar{v} \pm I_{95\%}$	1646.7 ± 0.2	1717.2 ± 0.9
	Validation	1647.3	1719.5

example, with wide time windows under small uncertainty and for $n_S = 1000$ scenarios, the average solving time increases by 40% when increasing λ from 0 to 0.01. Figure 7 summarizes the effect of uncertainty amplitude, IAF time-window width, and λ on the average CPU time. In terms of objective-function values, large differences are obvious between results with $\lambda = 0$ and those with $\lambda = 0.01$.

As expected, since the objective-function value with $\lambda = 0.01$ is increased by the term $0.01 \times \sum_{i \in \mathcal{A}} x_i$ (called *weighted total completion time*), it is not directly comparable to the objective-function value with $\lambda = 0$. However, we still can extract and compare separately the values of the three criteria: weighted total completion time, landing sequence length and second-stage cost, displayed on Figure 8 for narrow time windows at the IAF and small uncertainty. Although, the weighted total completion time is not included in the objective function with $\lambda = 0$, it was recomputed separately after the optimization and plotted on Figure 8. Figure 8 reveals that minimizing the sum of target times at the IAF in the first stage in addition to the landing sequence length has no effect on the obtained landing sequence length, whereas it decreases the weighted total completion time, as expected. However, this decrease comes at the expense of increasing the second-stage cost. Therefore, to save computing time and decrease the second-stage terminal-area impact

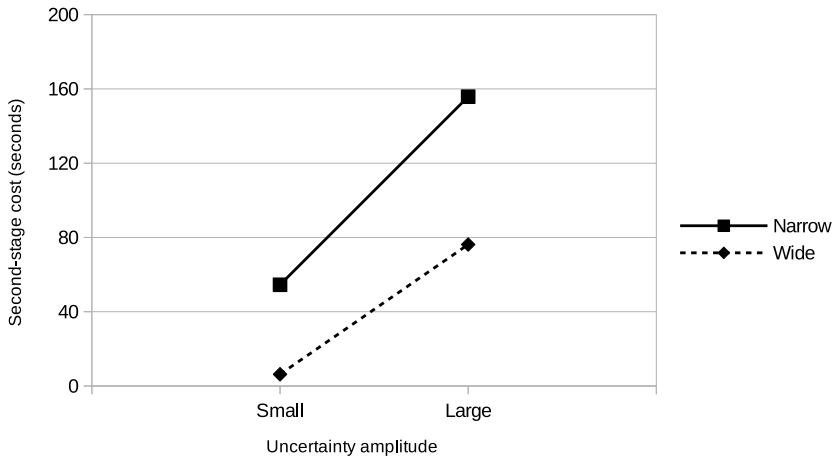


Fig. 6 Effect of uncertainty amplitude and IAF time-window width on the second-stage cost, for $\lambda = 0$ and $n_{\mathcal{S}} = 1000$.

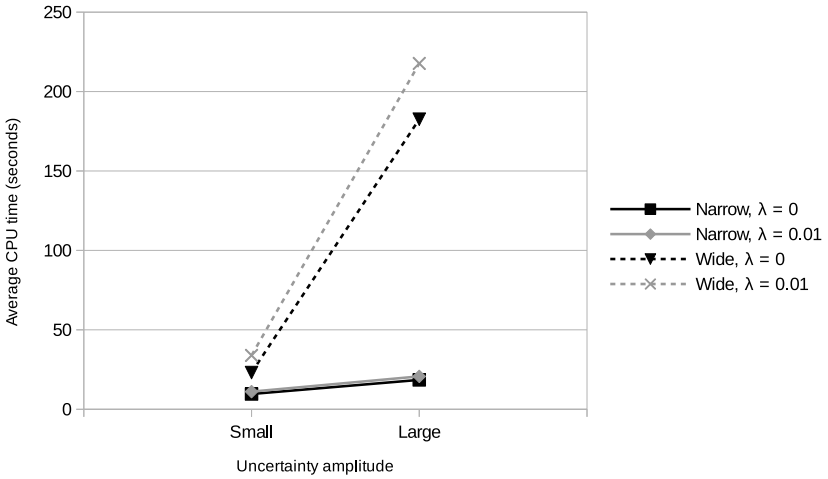


Fig. 7 Effect of uncertainty amplitude, IAF time-window width, and λ on the average CPU time for $n_{\mathcal{S}} = 200$.

cost, we shall in the sequel focus only on minimizing the landing sequence length in the first stage ($\lambda = 0$).

Table 7 Scenario-based approach results: instance 1, narrow IAF time windows, $\lambda = 0$

σ	Small	Large
$n_{\mathcal{S}}$	400	600
$n_{\mathcal{R}}$	1	1
CPU (s)	47.7 (+ 12.8)	228.2 (+ 13.3)
$v^\star$	940.7	1040.3
Validation	941.7	1043.2

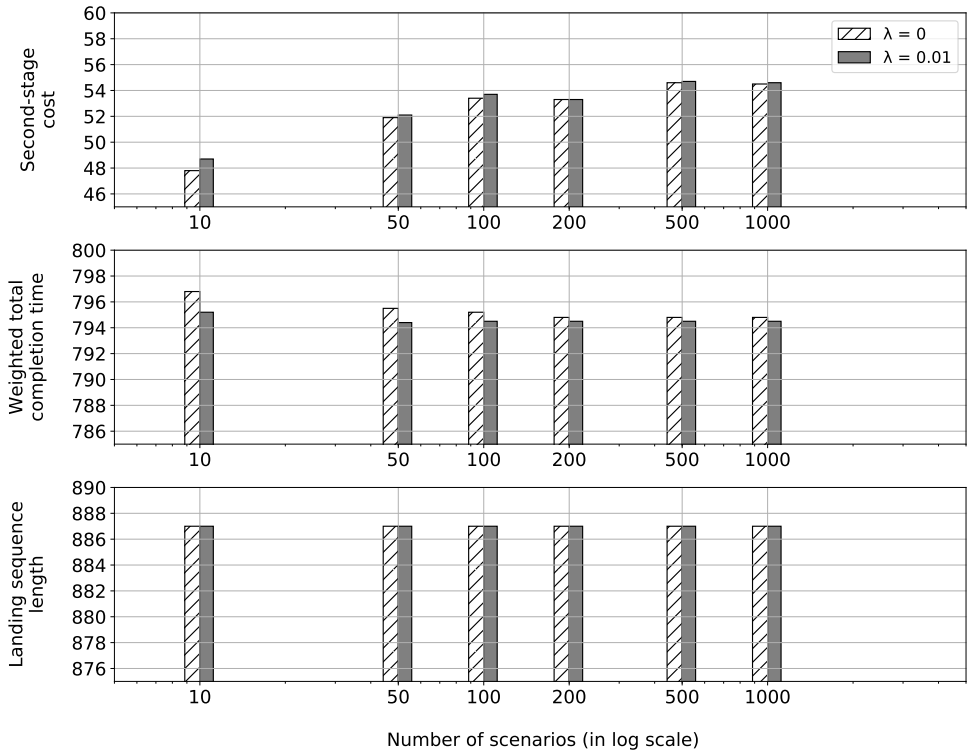


Fig. 8 Average landing sequence length, weighted total completion time and second-stage cost for instance 1, narrow IAF time windows, small uncertainty and $\lambda = 0$ and 0.01.

IV.D Comparison of real-time solution methods: scenario-based versus replication-based approaches

Based on the numerical study reported in the previous subsection, we retain the following two settings: narrow time windows at the IAF and minimizing only the landing sequence length in the first stage ($\lambda = 0$). Uncertainty is expected to be smaller in a shorter time horizon, as shown in [1, 11]. This leads us to consider, in a real-time context, a small CPU time budget for optimization under small uncertainty, and a relatively larger time budget under large uncertainty. More precisely, we consider 1 and 5 minutes as time budgets for small and large uncertainty, respectively.

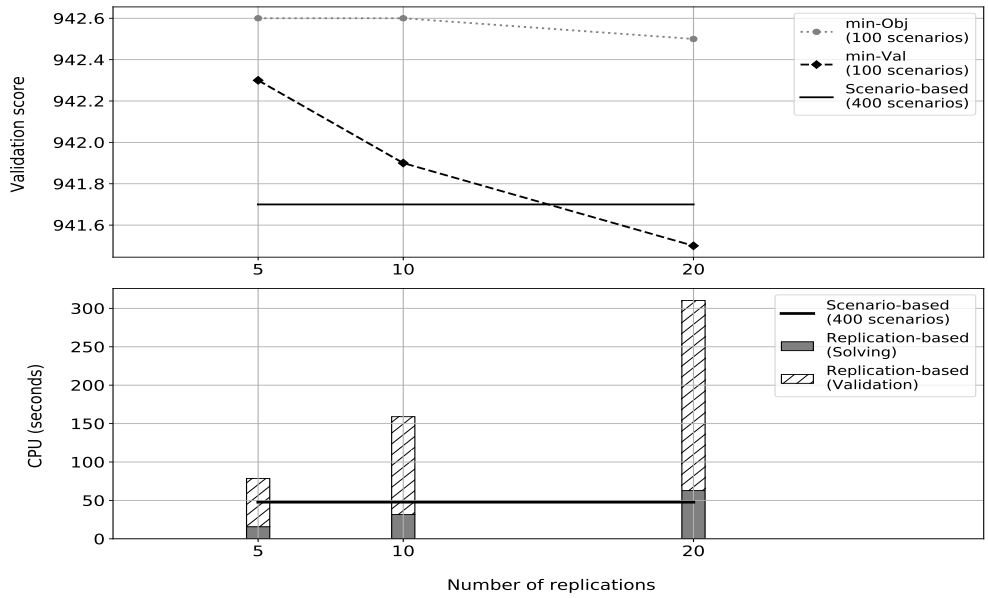
Table 8 Replication-based approach results: instance 1, narrow IAF time windows, $\lambda = 0$, small uncertainty

$n_{\mathcal{R}}$	solution	*min-Obj*	*min-Val*
5	CPU (s)	15.8 (+ 62.8)	
	$v^\star$	936.7	937.3
	Validation	942.6	942.3
10	CPU (s)	31.7 (+ 127.3)	
	$v^\star$	936.7	938.1
	Validation	942.6	941.9
20	CPU (s)	62.8 (+ 247.5)	
	$v^\star$	931.3	938.3
	Validation	942.5	941.5

Table 9 Replication-based approach results: instance 1, narrow IAF time windows, $\lambda = 0$, large uncertainty

$n_{\mathcal{R}}$	solution	*min-Obj*	*min-Val*
	CPU (s)	91.0 (+ 63.7)	
5	$v^\star$	1032.7	1036.3
	Validation	1043.7	1043.6
	CPU (s)	185.2 (+ 128.3)	
10	$v^\star$	1031.2	1031.2
	Validation	1043.5	1043.5
	CPU (s)	281.8 (+ 194.4)	
15	$v^\star$	1029.1	1044.6
	Validation	1045.2	1043.1

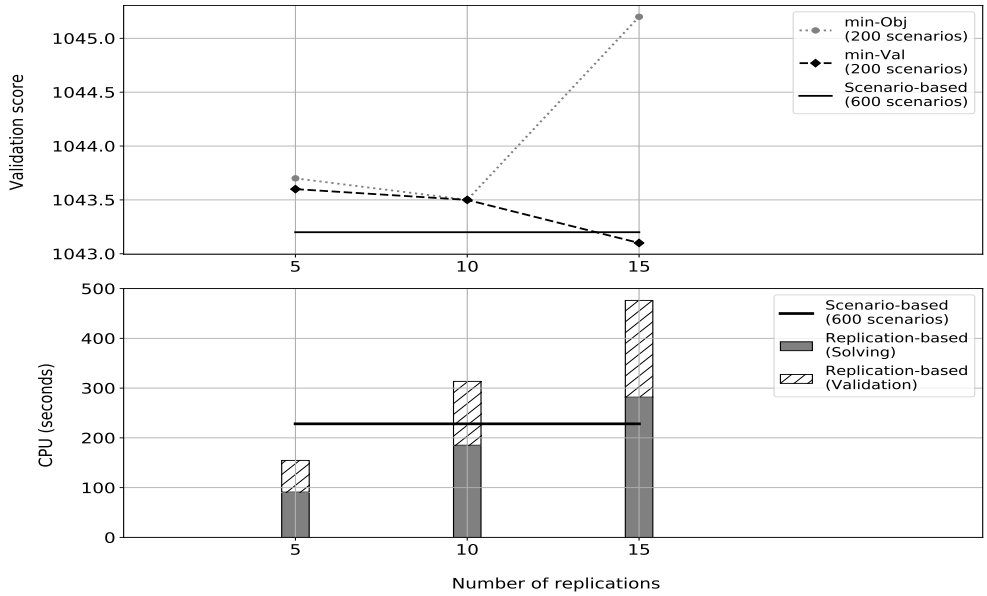
Fig. 9 Solution characteristics from replication-based approach under small uncertainty



In this subsection, we compare scenario-based and replication-based approaches, introduced in Section III, to solve the SAA problem. Given the average CPLEX solving time reported in Subsection IV.C, we select $n_{\mathcal{S}} = 100$ scenarios for the replication-based approach under small uncertainty and $n_{\mathcal{S}} = 200$ scenarios under large uncertainty, while the number of replications is adjusted to fit the time budget. In the scenario-based approach, only one SAA problem is solved with a relatively large number of scenarios (400 and 600 scenarios under small and large uncertainty, respectively).

On the one hand, a scenario-based approach naturally returns a unique solution. On the other hand, at the end of the solving process using a replication-based approach, we are left with a pool of solutions. To select a unique solution from this pool, we may either directly choose the minimum-objective-function-value solution (called *min-Obj*) as in [13], or re-evaluate all distinct solutions from the pool on a validation set and choose the minimum-validation-score solution (called *min-Val*) as in [11]. Results of the replication-based approach for both small and large uncertainties

Fig. 10 Solution characteristics from replication-based approach under large uncertainty



are summarized in Tables 8 and 9. Results of the scenario-based approach for both small and large uncertainty are summarized in Table 7. In these Tables, "CPU" stands for the solving time from CPLEX expressed in seconds, while the added time between brackets stands for the total validation time (for one solution from the scenario-based approach and for all solutions from the replication-based approach). For each retained solution, $v^\star$ represents the objective-function value, while "Validation" refers to its out-of-sample validation score, computed over $10,000$ scenarios. Figures 9 and 10 plot validation scores and computing times (solving and validation) for replication-based solutions (*min-Obj* and *min-Val*) in terms of the number of replications, under small and large uncertainty respectively. The performance of the scenario-based solution is shown in black thick line.

Regardless the uncertainty amplitude and the number of replications, the *min-Obj* solution clearly performs worse than the *min-Val* and the scenario-based solutions, as expected. Under small uncertainty, the scenario-based approach can be applied with a large number of scenarios ($n_{\mathcal{S}} = 400$) within the time budget of 1 minute. The *min-Val* solution from the replication-based approach for $n_{\mathcal{R}} = 20$ replications outperforms the scenario-based solution. However, it can only be obtained with around 5 minutes computing time. Consequently, if the time budget is limited to 1 minute, the scenario-based approach is recommended. Similar observations can be made for the case under large uncertainty.

As a conclusion, regardless the uncertainty amplitude, a fast solution method should be built on a scenario-based approach using a large enough number of scenarios, subject to the time budget. If a fast out-of-sample validation procedure is available, then any replication-based approach should follow the validation-score criterion to select a solution from the solution pool, while the minimum-objective-value-function criterion should be avoided.

IV.E Results for instance 2

As narrow time windows at the IAF and $\lambda = 0$ were identified as a suitable setting for instance 1 (n =10 aircraft), we keep them for our tests on instance 2 (n =14 aircraft). For each uncertainty amplitude, small and large, we solve a single SAA problem with 10 minutes as a time limit for CPLEX. We use 100 scenarios for small uncertainty and 200 scenarios for large uncertainty. Out-of-sample validation is performed using 10,000 scenarios. Results under small and large uncertainties, given in Table 10, show that instance 2 is much harder to solve than instance 1. Although CPLEX is unable to prove optimality within 10 minutes for 200 scenarios under large uncertainty, validation scores are not dramatically larger than the objective-function values. Nevertheless, an efficient solving algorithm is clearly needed to handle large numbers of aircraft.

Table 10 Scenario-based approach results: instance 2, narrow IAF time windows, and $\lambda = 0$

σ	Small	Large
$n_{\mathcal{S}}$	100	200
CPU (s)	444.1	Tilim
Status (Gap)	Opt. (0.0%)	Feas. (19.9%)
$v^\star$	1292.3	1513.2
Validation	1297.6	1519.6

IV.F Effect on FCFS policy performance in the terminal area

In this subsection, we consider a complete sequencing-and-scheduling process of aircraft arrivals from the time they are captured (two to three hours before landing) to the time they land. We consider two sequencing-and-scheduling points: the IAF, and the runway threshold. Firstly, captured aircraft are sequenced and scheduled for the IAF according to some policy, called the *pre-IAF sequencing-and-scheduling policy*. Due to uncertainties, actual aircraft IAF times are different from the IAF target times. Subsequently, when aircraft actually arrive at the IAF, some other policy is applied in order to sequence and schedule approaching aircraft to the runway threshold. Here, we assume that ATCs in the terminal area sequence aircraft for landing in the same order in which they actually pass over the IAF. This air traffic control policy will be referred to as *FCFS policy in the terminal area*. We aim at evaluating the effect of our retained solutions from fast solution methods, reported in Table 7, on the expected performance of the FCFS policy in the terminal area. For that purpose, we compare the expected performance of the FCFS policy in the terminal area in different upstream situations i.e, when different pre-IAF sequencing-and-scheduling policies are applied. The baseline upstream situation consists in scheduling aircraft at the IAF in a FCFS fashion according to their planned times at the IAF. The minimum time separation enforced over the IAF is simply $\underline{S}^I$ (72 seconds). This baseline pre-IAF sequencing-and-scheduling policy is called the *pre-IAF FCFS policy*. Enforcing a minimum separation between aircraft that is larger than operational requirements is a common technique to hedge against uncertainty [9]. When the pre-IAF FCFS policy is applied with an

enlarged minimum separation over the IAF, we call it a pre-IAF *buffered* FCFS policy. Since the minimum separation over the IAF is commonly expressed in NM, we consider two values for the distance buffer, 1 NM and 2 NM, resulting in time buffers of 14 and 28 seconds respectively. The two resulting pre-IAF sequencing-and-scheduling policies are called pre-IAF 1-buffered FCFS policy and pre-IAF 2-buffered FCFS policy. These last two policies define the second and the third upstream situations respectively. The fourth upstream situation relies on our stochastic-optimization approach to pre-sequence and pre-schedule aircraft over the IAF. The four pre-IAF policies will be noted "FCFS-0", "FCFS-1", "FCFS-2", and "StochOpt" respectively. For each upstream situation, 10,000 scenarios are simulated and various performance measures of the FCFS policy in the terminal area, described below, are computed.

Performance measures:

- Average number of conflicts at the IAF, noted "IAF conflicts", computed as the average number of separation violations over the IAF between any pair of aircraft
- Average landing rate per hour, noted "landing rate"
- Average last landing time, noted "last landing"
- Average total time-to-lose in the terminal area, noted "TMA total time-to-lose", the time-to-lose for a single aircraft in the terminal area being computed as the (positive) deviation from the aircraft target landing time with respect to its unconstrained landing time
- Average maximum time-to-lose in the terminal area, noted "TMA max time-to-lose"

Tables 11 and 12 report values of these performance measures respectively for instance 1 and a compressed version of instance 1, under the four upstream situations. The modified version of instance 1 was obtained by compressing the planned IAF schedule of instance 1 by a factor two. Hence, the time span over the IAF is contracted from 6:00 - 6:20 AM to 6:00 - 6:10 AM, while the number of aircraft, $n = 10$, is conserved. We computed our solutions for the compressed version of instance 1 using $n_S = 100$ scenarios for both small and large uncertainties. Solving times are respectively 59.1 and 71.6 seconds. The target sequence and IAF target times, for each pre-IAF policy, are illustrated in Figures 11 and 12 for the original instance 1 and its compressed version, respectively. In each of the two figures, "StochOpt-30" and "StochOpt-60" refer to our stochastic programming policy under small and large uncertainty respectively. IAF target times of a given aircraft that has the same position in the sequence across the different pre-IAF policies are linked with a continuous line, while dashed lines are used for aircraft whose position changes at least under one pre-IAF policy.

In moderate-density traffic, as in instance 1 (Table 11), solutions from our stochastic-optimization approach decrease the expected number of conflicts over the IAF, *e.g*, down to -70% under small uncertainty, and the time-to-lose within the terminal area, *e.g*, down to -86% in terms of total time-to-lose in the terminal area under small uncertainty, compared to the pre-IAF FCFS policy. On the other hand, average landing rates and last landing times using our approach are

Table 11 FCFS performance in the terminal area: instance 1 with narrow IAF time windows

uncertainty	pre-IAF policy	IAF conflicts	landing rate	last landing	TMA time-to-lose total	TMA time-to-lose max
Small	FCFS - 0	3.1	**26.9**	**06:33:29**	8 min 23 s	2 min 35 s
	FCFS - 1	2.2	26.8	06:33:35	6 min 01 s	2 min 01 s
	FCFS - 2	1.5	26.6	06:33:43	4 min 09 s	1 min 35 s
	StochOpt	**0.9**	25.0	06:34:13	**1 min 08 s**	**0 min 41 s**
Large	FCFS - 0	3.6	**26.9**	**06:33:42**	9 min 02 s	2 min 48 s
	FCFS - 1	3.0	26.6	06:33:55	7 min 25 s	2 min 24 s
	FCFS - 2	2.5	26.3	06:34:11	6 min 11 s	2 min 07 s
	StochOpt	**1.8**	24.7	06:34:39	**2 min 47 s**	**1 min 22 s**

Table 12 FCFS performance in the terminal area: compressed instance 1 with narrow IAF time windows

uncertainty	pre-IAF policy	IAF conflicts	landing rate	last landing	TMA time-to-lose total	TMA time-to-lose max
Small	FCFS - 0	3.9	34.8	06:28:24	16 min 09 s	4 min 20 s
	FCFS - 1	2.7	34.6	06:28:32	10 min 24 s	2 min 51 s
	FCFS - 2	**1.8**	33.9	06:28:52	5 min 35 s	1 min 44 s
	StochOpt	2.9	**37.5**	**06:26:10**	**5 min 10 s**	**1 min 24 s**
Large	FCFS - 0	5.0	34.6	06:28:42	18 min 05 s	4 min 27 s
	FCFS - 1	3.9	34.1	06:28:57	12 min 51 s	3 min 14 s
	FCFS - 2	3.2	33.2	06:29:25	8 min 31 s	2 min 23 s
	StochOpt	**2.8**	**37.5**	**06:26:10**	**5 min 12 s**	**1 min 24 s**

slightly worse than the pre-IAF FCFS policy. This may be due to the fact that our stochastically-optimized schedules at the IAF are too sparse (see Figure 11). With regard to the two pre-IAF buffered FCFS policies with 1 NM or 2 NM, they can be an interesting compromise in moderate traffic since they perform close to our approach while impacting less the landing rate and the average last landing time.

In high-density traffic, as in the compressed instance 1 (Table 12), our stochastic sequencing-and-scheduling policy outperforms the pre-IAF FCFS policy almost in all measures. In terms of expected number of IAF conflicts, under large uncertainty, 44% less separation violations are likely to occur on average compared to the unbuffered pre-IAF FCFS policy. The time-to-lose inside the terminal area dramatically decreases as well, *e.g*, down to -71% in terms of total time-to-lose in the terminal area under large uncertainty. Also, the average last landing time is earlier by more than 2 minutes, which increases the average landing rate, *e.g*, by 7.7% under large uncertainty. In high-density traffic, pre-IAF buffered FCFS policies may efficiently decrease the expected number of IAF conflicts and the time-to-lose inside the

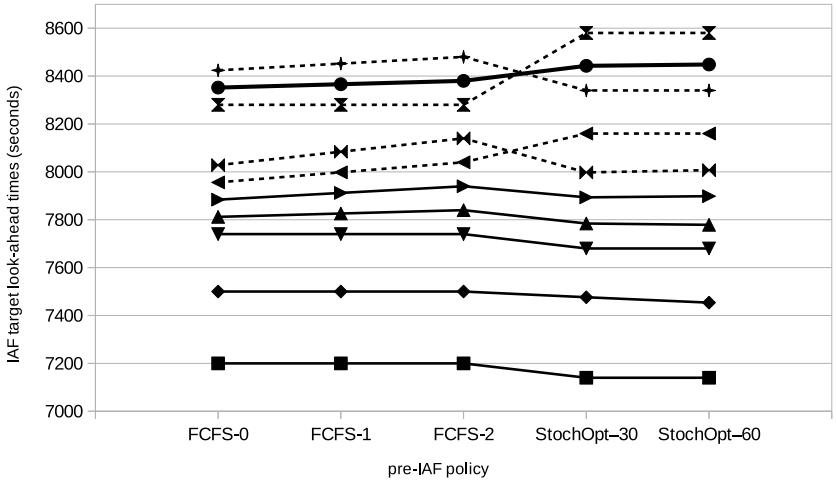


Fig. 11 IAF target times and sequences for instance 1 with different pre-IAF policies.

terminal area. However, average landing rates slightly decline, most likely because of too sparse IAF target times (due to the enlarged IAF separation requirements).

To study further the effect of uncertainty, we test our approach with a standard deviation of $\sigma = 200$ seconds. Remark that under this "very large" uncertainty, more than 93% of random IAF time deviations fall within ±5 minutes and more than 99% within ±10 minutes. Results in moderate-density traffic (instance 1) confirm the trend observed under small and large uncertainties. In high-density traffic (compressed instance 1), average landing rates are maintained compared to the unbuffered pre-IAF FCFS policy, while significant improvements are made on all of the remaining performance measures.

Finally, recall that a single tour in a holding stack is usually flown in 4 minutes. Measures of time-to-lose in the terminal area, especially in high-density traffic, show that our stochastic-optimization approach may avoid resorting to such holding stacks. Under a FCFS policy in the terminal area, we may conclude that our optimization approach over the IAF successfully transforms a *circular holding* (holding patterns at the entry of the terminal area) into a linear holding applied when aircraft are still a few hours away from landing, while increasing the landing rate, as expected from an efficient E-AMAN.

V Conclusions

In this paper, we have presented a computational study on the problem of sequencing and scheduling arrivals over a single IAF under uncertain times at the IAF. Such a problem is relevant due to the foreseen extension of AMAN operational horizon up to a few hours before landing. We relied on a two-stage stochastic programming approach exploiting CPLEX solver. The SAA method was used to make the problem tractable and to find satisfying approximate solutions. The effect of different problem characteristics (narrow vs. wide time windows at the IAF, small

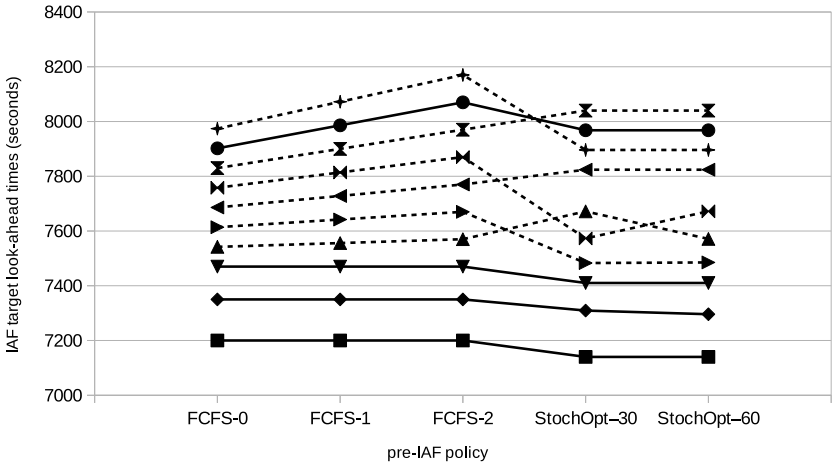


Fig. 12 IAF target times and sequences for the compressed version of instance 1 with different pre-IAF policies.

vs. large uncertainty) as well as different optimization parameters (first-stage objective function, number of second-stage scenarios) were analyzed in order to evaluate the viability of our approach. Scenario-based and replication-based approaches were tested and compared as potential solution methods for real-time implementation. Realistic instances involving 10 and 14 arrivals on CDG were used as a case study. For 10 aircraft with narrow time windows at the IAF, our approach returns good-quality solutions in a short solving time (less than 1 minute for small uncertainty and less than 5 minutes for large uncertainty). Simulation-based validation experiments show that our retained solutions can decrease the number of expected conflicts over the IAF by more than 70% and the total time-to-lose inside the terminal area by 86% over a FCFS policy, at the expense of a slight decrease of the landing rate in moderate-density traffic under small uncertainty. In high-density traffic, besides alleviating traffic complexity near the terminal area, our solutions enhance the expected landing rate by more than 7% under large uncertainty.

Acknowledgements

The authors would like to thank Serge Roux (ENAC) for providing the arrivals data on Paris Charles-De-Gaulle (CDG) airport. F. Bastin and B. Gendron are funded by Canada NSERC Discovery Grants.

References

[1] Tielrooij, M., Borst, C., Van Paassen, M. M., and Mulder, M., "Predicting arrival time uncertainty from actual flight information," *Proceedings of the 11th USA/Europe Air Traffic Management Research and Development Seminar*, 2015, pp. 577–586.

[2] Dear, R. G., "The dynamic scheduling of aircraft in the near terminal area," Tech. rep., Massachusetts Institute of Technology, 1976.

[3] Bennell, J. A., Mesgarpour, M., and Potts, C. N., "Airport runway scheduling," *4OR*, Vol. 9, No. 2, 2011, pp. 115–138.

doi:10.1007/s10288-011-0172-x.

[4] Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., and Abramson, D., "Scheduling aircraft landings: The static case," *Transportation Science*, Vol. 34, No. 2, 2000, pp. 180–197. doi:10.1287/trsc.34.2.180.12302.

[5] Ghoniem, A., Farhadi, F., and Reihaneh, M., "An accelerated branch-and-price algorithm for multiple-runway aircraft sequencing problems," *European Journal of Operational Research*, Vol. 246, No. 1, 2015, pp. 34–43. doi:10.1016/j.ejor.2015.04.019.

[6] Lieder, A., and Stolletz, R., "Scheduling aircraft take-offs and landings on interdependent and heterogeneous runways," *Transportation Research Part E*, Vol. 88, 2016, pp. 167–188. doi:10.1016/j.tre.2016.01.015.

[7] Balakrishnan, H., and Chandran, B. G., "Algorithms for scheduling runway operations under constrained position shifting," *Operations Research*, Vol. 58, No. 6, 2010, pp. 1650–1665. doi:10.1287/opre.1100.0869.

[8] Bennell, J. A., Mesgarpour, M., and Potts, C. N., "Dynamic scheduling of aircraft landings," *European Journal of Operational Research*, Vol. 258, No. 1, 2017, pp. 315–327. doi:10.1016/j.ejor.2016.08.015.

[9] Meyn, L. A., and Erzberger, H., "Airport arrival capacity benefits due to improved scheduling accuracy," *Proceedings of the 5th Aviation, Technology Integration and Operations and the 16th Lighter-Than-Air Systems Technology and Balloon Systems Conferences*, 2005.

[10] Lee, H., and Balakrishnan, H., "A study of tradeoffs in scheduling terminal-area operations," *Proceedings of the IEEE*, Vol. 96, No. 12, 2008, pp. 2081–2095. doi:10.1109/JPROC.2008.2006145.

[11] Sölveling, G., Solak, S., Clarke, J.-P., and Johnson, E., "Runway operations optimization in the presence of uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 5, 2011, pp. 1373–1382. doi:10.2514/6.2010-9252.

[12] Sölveling, G., and Clarke, J.-P., "Scheduling of airport runway operations using stochastic branch and bound methods," *Transportation Research Part C*, Vol. 45, 2014, pp. 119–137. doi:10.1016/j.trc.2014.02.021.

[13] Bosson, C. S., and Sun, D., "Optimization of Airport Surface Operations Under Uncertainty," *Journal of Air Transportation*, Vol. 24, No. 3, 2016, pp. 84–92. doi:10.2514/1.D0013.

[14] Heidt, A., Helmke, H., Kapolke, M., Liers, F., and Martin, A., "Robust runway scheduling under uncertain conditions," *Journal of Air Transport Management Part A*, Vol. 56, 2016, pp. 28–37. doi:10.1016/j.jairtraman.2016.02.009.

[15] Heidt, A., "Uncertainty Models for Optimal and Robust ATM Schedules," Ph.D. thesis, Friedrich Alexander University, Erlangen, Germany, 2017.

[16] Kapolke, M., Fürstenau, N., Heidt, A., Liers, F., Mittendorf, M., and Weiß, C., "Pre-tactical optimization of runway utilization under uncertainty," *Journal of Air Transport Management Part A*, Vol. 56, 2016, pp. 48–56. doi:10.1016/j.jairtraman.2016.02.004.

[17] Frankovich, M. J., "Air traffic flow management at airports: a unified optimization approach," Ph.D. thesis, Massachusetts Institute of Technology, 2012.

[18] Khassiba, A., Bastin, F., Cafieri, S., Gendron, B., and Mongeau, M., "Extended aircraft arrival management under uncertainty: A chance-constrained two-stage mixed-integer programming model," Tech. Rep. 2018-55, CIRRELT, Université de Montréal, Canada, 2018.

[19] Shapiro, A., and Homem-de Mello, T., "On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs," *SIAM Journal on Optimization*, Vol. 11, No. 1, 2000, pp. 70–86. doi:10.1137/S1052623498349541.

[20] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B., "Julia: A Fresh Approach to Numerical Computing," *SIAM Review*, Vol. 59, No. 1, 2017, pp. 65–98. doi:10.1137/141000671.

[21] Xue, M., and Zelinski, S., "A stochastic scheduler for integrated arrival, departure, and surface operations in Los Angeles," *Proceedings of the 15th Aviation Technology, Integration, and Operations Conference*, AIAA, 2015. doi:10.2514/6.2015-2273.