Extended Aircraft Arrival Management under Uncertainty: A Computational Study

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

CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Extended Aircraft Arrival Management under Uncertainty: A Computational Study

Ahmed Khassiba
Fabian Bastin
Bernard Gendron
Sonia Cafieri
Marcel Mongeau

December 2018

CIRRELT-2018-52

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Extended Aircraft Arrival Management under Uncertainty: A Computational Study

Ahmed Khassiba[1,2,3,*], Fabian Bastin[1,2], Bernard Gendron[1,2], Sonia Cafieri[3], Marcel Mongeau[3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[3] École Nationale de l'Aviation Civile (ENAC), 7, avenue Edouard Belin, CS 54005, 31055 Toulouse Cedex 4, France

Abstract. Arrival Manager (AMAN) 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 (IAF) 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 90%. Moreover, assuming that air traffic controllers (ATCs) schedule landings in a first-come first-served order, the ATC expected workload in the terminal area can be decreased by more than 98%, while the expected landing rate remains unchanged. 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.

Keywords. Aircraft arrival scheduling, uncertainty, computational study.

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 the Natural Sciences and Engineering Research Council of Canada (NSERC)by Canada NSERC Discovery Grants.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

[*] Corresponding author: Ahmed.Khassiba@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2018

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 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 the scheduling horizons are extended, uncertainties about predicted times of arrival get larger and frequent deterministic re-optimization becomes useless. Hence, uncertainties need to be embedded within the optimization model for 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 (ASS). 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 ASS: 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 ASS 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 the expected ATC workload 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 and perspectives 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 the expected ATC workload 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.

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.

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

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

Table 1 Final-approach separations (seconds) at CDG according to ICAO wake-turbulence categories

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

Besides anticipating the optimal landing sequence, making aircraft arrive as early as possible at the IAF might help maximizing the landing rate. For that reason, we may try to minimize $\sum_{i \in \mathcal{A}} x_i$. Overall, we seek to maximize the expected landing rate through minimizing 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:

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

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. In this second-stage problem, we schedule aircraft to the runway threshold so as to minimize the ATC workload in the terminal area.

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. For each aircraft $i \in \mathcal{A}$, U_i is computed as the sum of its actual arrival time at the IAF and the *unconstrained flight time* from the IAF to the runway threshold. This unconstrained landing time is defined so that controlling aircraft i in the terminal area to land at U_i corresponds to the minimum workload for ATCs with regard to aircraft i. Hence, $|y_i - U_i|$ will quantify the additional ATC workload required by aircraft i in the terminal area. Here, we assume that expediting or delaying an aircraft by the same amount of time would generate the same additional ATC workload. We define a second-stage scenario s as a possible realization of actual arrival times at the IAF of all aircraft in $\mathcal{A}$. Remark that unconstrained landing times depend on the scenario s. Hence, unconstrained landing time of aircraft i in scenario s will be noted U_i^s.

Therefore, given a scenario s, the second-stage problem reads:

$$\begin{aligned} Q(x,\delta,s) := \min_{y} \sum_{i\in\mathcal{A}} \left| y_i - U_i^s \right| & \\ \text{s.t. final approach separations} & \\ \text{and time-windows constraints for landing} & \end{aligned}$$

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:

$$v^\star \quad := \quad \min_{x,\delta} \quad f_1(x,\delta) + \mathbb{E}\left[Q(x,\delta,.)\right]$$

where $\mathbb{E}[.]$ is the expectation operator. 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 ATC workload in the terminal area. 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 [17] 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:

$$\hat{v}(\mathcal{S}) := \quad \min_{x,\delta} \quad f_1(x,\delta) + \frac{1}{n_{\mathcal{S}}}\sum_{s\in\mathcal{S}} Q(x,\delta,s)$$

Since $\hat{v}(\mathcal{S})$ depends on the scenarios set $\mathcal{S}$, $\hat{v}(\mathcal{S})$ is, itself, a random variable. The SAA method [17] 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 $\hat{v}\left(\mathcal{S}\right)$ 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 $|\mathcal{S}_r| = n_{\mathcal{S}}$:

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

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, Solveling 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 [18] 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.

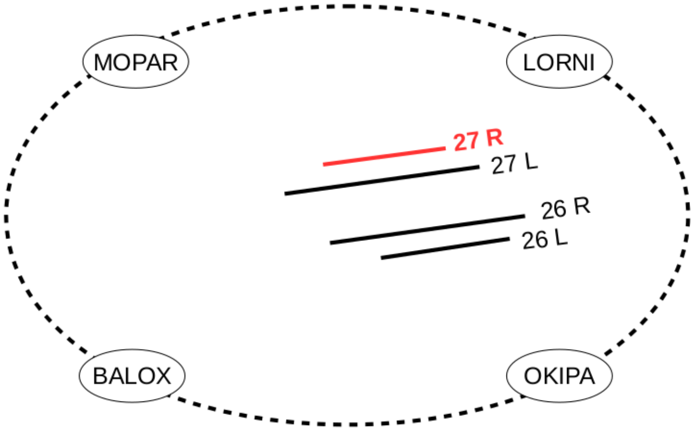
The two instances and the different problem characteristics are described in Subsection IV.A. The optimization parameters along with their tested values are presented in Subsection IV.B. The main results for the first instance are reported and discussed in Subsection IV.C. To select an effective real-time 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.

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

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



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. The tested time windows are: [-5 min, +5 min] (narrow) and [-5 min, +15 min] (wide).

Uncertainty

Following the literature ([9, 19]), 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 in both instances, we opt for wide time windows for landing, i.e. [-5 min, +15 min], and 15 minutes as an unconstrained flight time from the IAF to the runway threshold.

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. For each setting combination, $n_{\mathcal{R}} = 10$ replications are performed. Far each replication, the time limit in CPLEX solver is set to 2 hours. 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. “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 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 $\bar{v}\left(n_{\mathcal{S}}, n_{\mathcal{R}}\right)$ in Section III. For each computed value of $\bar{v}\left(n_{\mathcal{S}}, n_{\mathcal{R}}\right)$, 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 $\mathbb{E}\left[\hat{v}\left(\mathcal{S}\right)\right]$ lies within this interval.

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

	TW^I	Narrow	
$n_{\mathcal{S}}$	σ	Small	Large
10	CPU (s)	3.6	11.2
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	828.0 ± 2.4	892.7 ± 16.3
	Validation	844.3	939.1
50	CPU (s)	10.9	62.4
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	832.9 ± 1.4	906.7 ± 5.6
	Validation	837.3	920.8
100	CPU (s)	28.7	165.4
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	833.9 ± 1.0	910.6 ± 2.3
	Validation	837.1	919.1
200	CPU (s)	52.9	410.1
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	834.8 ± 0.8	911.9 ± 3.0
	Validation	836.8	918.6
500	CPU (s)	72.6	2947.9
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	835.5 ± 0.2	914.5 ± 1.6
	Validation	836.3	916.9
1000	CPU (s)	167.2	7200.0
	Status (Gap)	Opt. (0.0%)	Feas.(7.4%)
	$\bar{v} \pm I_{95\%}$	835.5 ± 0.2	914.5 ± 1.0
	Validation	836.2	916.6

"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 $n_{\mathcal{S}}$ 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, $n_{\mathcal{S}}$

As expected, the solving time increases rapidly with $n_{\mathcal{S}}$. In 6 cases out of 8, CPLEX is not able to prove optimality within 2 hours when $n_{\mathcal{S}}$ reaches 1000 scenarios. The average objective-function value, $\bar{v}$, clearly increases with the number of scenarios. As long as optimal solutions are found (Figures 2 and 3), validation scores reveal that increasing the number of scenarios lead to better-quality solutions. However, a non-optimal solution found with a large number of scenarios can be of worse quality than an optimal solution found with a smaller number of scenarios (Figures 4 and 5).

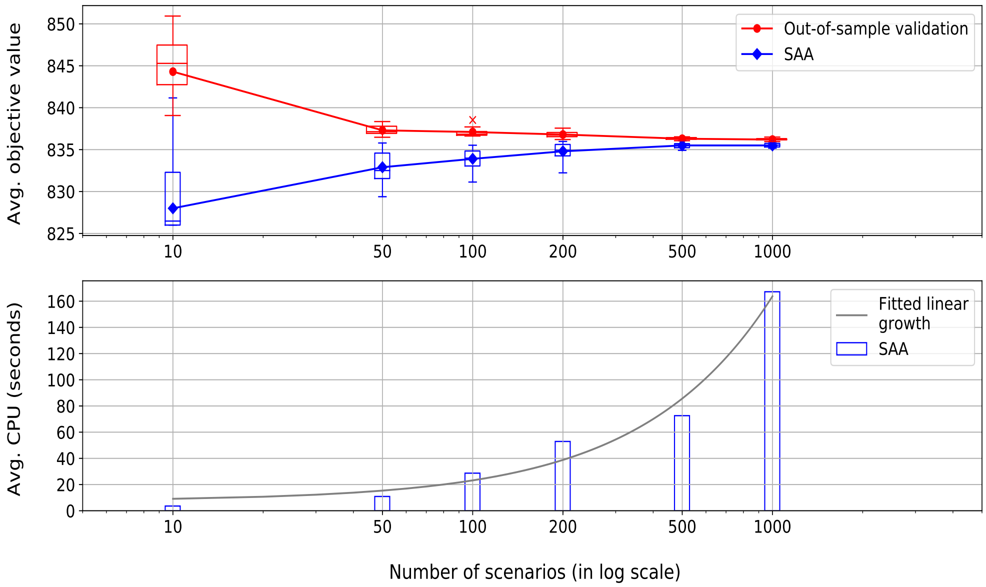


Fig. 2 Average objective-function values and validation scores (upper figure) and average CPU times (lower figure) for instance 1 with narrow IAF time windows, $\lambda = 0$ and small uncertainty.

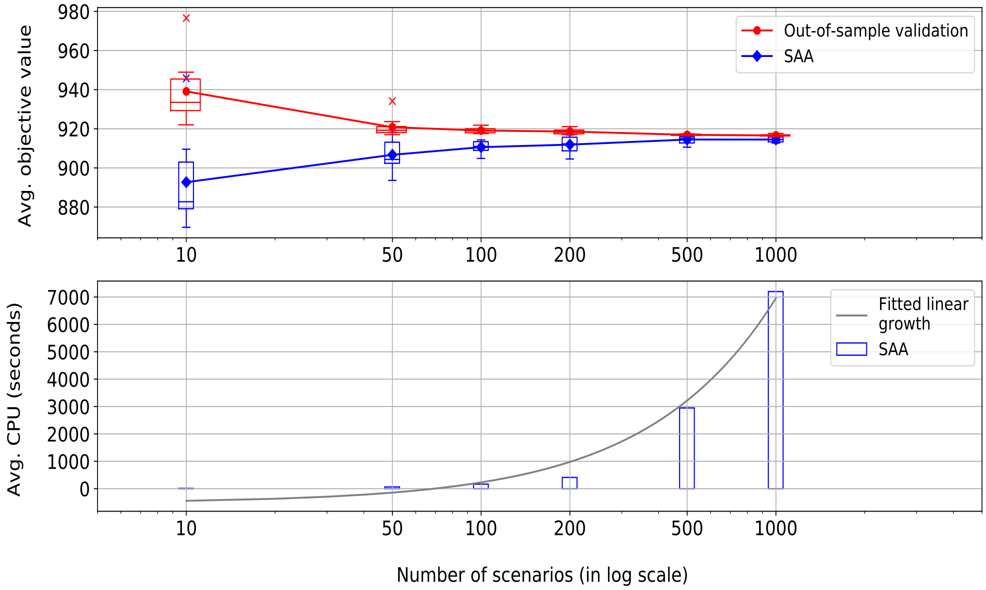


Fig. 3 Average objective-function values and validation scores (upper figure) and average CPU times (lower figure) for instance 1 with narrow IAF time windows, $\lambda = 0$ and large uncertainty.

Effect of time-windows 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 wide of the time windows. Consequently, there are many more feasible sequences to be considered.

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)	1.8	11.0
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1614.1 ± 3.0	1680.3 ± 15.7
	Validation	1633.6	1726.5
50	CPU (s)	6.1	51.3
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1620.3 ± 1.5	1694.2 ± 5.5
	Validation	1624.9	1707.4
100	CPU (s)	12.6	161.1
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1621.4 ± 1.1	1698.3 ± 2.4
	Validation	1624.4	1707.3
200	CPU (s)	26.2	436.6
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1622.4 ± 1.0	1699.7 ± 3.2
	Validation	1624.2	1706.7
500	CPU (s)	76.3	2579.9
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1622.9 ± 0.3	1702.2 ± 1.6
	Validation	1623.9	1704.5
1000	CPU (s)	155.5	7200.1
	Status (Gap)	Opt. (0.0%)	Feas.(4.0%)
	$\bar{v} \pm I_{95\%}$	1623.1 ± 0.2	1702.2 ± 1.0
	Validation	1623.7	1704.3

However, these additional sequences imply more delay for the aircraft. Therefore, the increased flexibility offered by wide time windows does not imply significantly better values of the objective function, as shown in the Tables 3 and 5 for example. Finally, in our context, narrow time windows are also preferred to wide ones because they are likely to boost aircraft on-time performance.

Effect of uncertainty amplitude

Throughout all the results, as uncertainty gets larger, more time is needed to solve the problem. For runs with narrow time windows and $n_S = 100$ scenarios, the average CPU time may increase more than 10 times when the uncertainty standard deviation is doubled. The increase factor is even further doubled when we switch from narrow to wide time windows. Therefore, with wide time windows, the solving time is more sensitive to uncertainty.

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)	18.9	484.8
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	828.2 ± 3.3	877.3 ± 10.0
	Validation	842.9	912.3
50	CPU (s)	107.0	3891.5
	Status (Gap)	Opt. (0.0%)	8 Opt. (1.1%)
	$\bar{v} \pm I_{95\%}$	827.6 ± 2.2	885.6 ± 10.1
	Validation	831.7	889.2
100	CPU (s)	267.2	5682.8
	Status (Gap)	Opt. (0.0%)	5 Opt. (4.0%)
	$\bar{v} \pm I_{95\%}$	828.7 ± 1.3	889.3 ± 10.9
	Validation	830.4	885.9
200	CPU (s)	2564.8	7200.0
	Status (Gap)	Opt. (0.0%)	Feas.(11.0%)
	$\bar{v} \pm I_{95\%}$	836.3 ± 3.3	904.6 ± 13.7
	Validation	835.1	901.3
500	CPU (s)	6233.0	7200.1
	Status (Gap)	4 Opt. (1.8%)	Feas.(14.3%)
	$\bar{v} \pm I_{95\%}$	836.8 ± 3.9	904.4 ± 9.6
	Validation	836.1	897.3
1000	CPU (s)	7200.1	7200.2
	Status (Gap)	Feas.(7.9%)	Feas.(16.4%)
	$\bar{v} \pm I_{95\%}$	840.9 ± 4.3	926.3 ± 18.7
	Validation	838.6	919.4

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

In terms of solving time, no significant effect is clear when minimizing the sum of target times at the IAF in addition to the landing sequence length in the first stage. Throughout the results in Tables 3 to 6, we remark that the average CPLEX solving time with $\lambda = 0$ and $\lambda = 0.01$ are close, contrary to the objective-function values. 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 makespan*), 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 makespan, landing sequence length and second-stage cost, displayed on Figure 6 for narrow time windows at the IAF and small uncertainty. Although, the weighted makespan is not included in the objective function with $\lambda = 0$, it was recomputed separately after the optimization and plotted on Figure 6. Figure 6 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 makespan, as expected. However, this decrease comes at the expense of increasing the expected workload in the terminal area (second-stage cost). Therefore, to save computing time and decrease ATC workload, we shall in the sequel focus only on minimizing

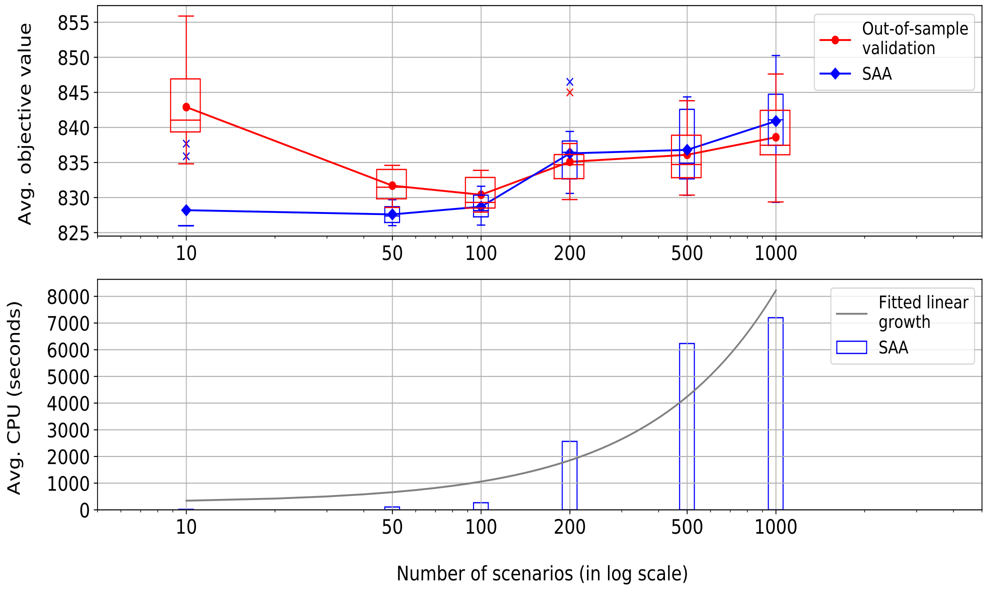


Fig. 4 Average objective-function values and validation scores (upper figure) and average CPU times (lower figure) for instance 1 with wide IAF time windows, $\lambda = 0$, small uncertainty.

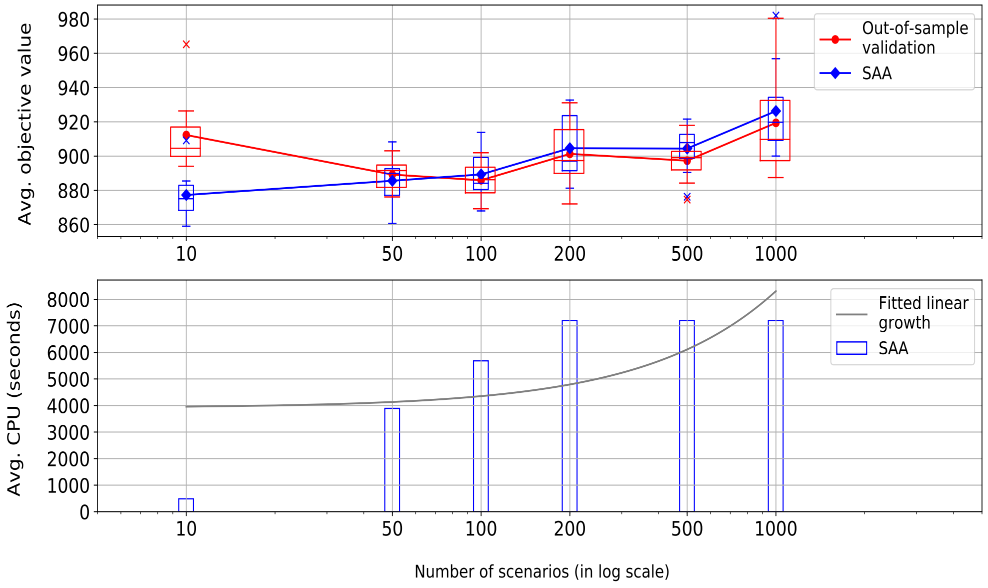


Fig. 5 Average objective-function values and validation scores (upper figure) and average CPU times (lower figure) for instance 1 with wide IAF time windows, $\lambda = 0$, large uncertainty.

the landing sequence length in the first stage ($\lambda = 0$).

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

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)	18.9	224.1
	Status (Gap)	Opt. (0.0%)	Opt. (0.0%)
	$\bar{v} \pm I_{95\%}$	1612.8 ± 2.2	1661.0 ± 12.7
	Validation	1635.3	1702.0
50	CPU (s)	98.2	4382.7
	Status (Gap)	Opt. (0.0%)	8 Opt. (0.9%)
	$\bar{v} \pm I_{95\%}$	1620.6 ± 2.9	1694.5 ± 9.2
	Validation	1626.2	1695.4
100	CPU (s)	258.1	6763.3
	Status (Gap)	Opt. (0.0%)	1 Opt. (4.1%)
	$\bar{v} \pm I_{95\%}$	1621.3 ± 1.1	1698.6 ± 10.9
	Validation	1624.8	1697.3
200	CPU (s)	1155.3	7200.0
	Status (Gap)	Opt. (0.0%)	Feas.(5.0%)
	$\bar{v} \pm I_{95\%}$	1625.9 ± 4.9	1689.7 ± 5.0
	Validation	1627.2	1686.7
500	CPU (s)	5569.5	7200.3
	Status (Gap)	6 Opt. (0.8%)	Feas. (8.1%)
	$\bar{v} \pm I_{95\%}$	1627.4 ± 6.1	1702.7 ± 9.0
	Validation	1626.6	1693.3
1000	CPU (s)	7200.0	7200.4
	Status (Gap)	Feas.(3.2%)	Feas. (8.4%)
	$\bar{v} \pm I_{95\%}$	1627.8 ± 5.0	1693.1 ± 7.4
	Validation	1627.2	1689.2

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 4 and 10 minutes for small and large uncertainty, respectively.

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 50 scenarios for the replication-based approach, 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 (1000 and 250 scenarios under

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

σ	Small	Large
n_S	1000	250
CPU (s)	203.1 (+ 11.9)	609.4 (+ 12.8)
$v^\star$	835.5	916.3
Validation	836.0	918.0

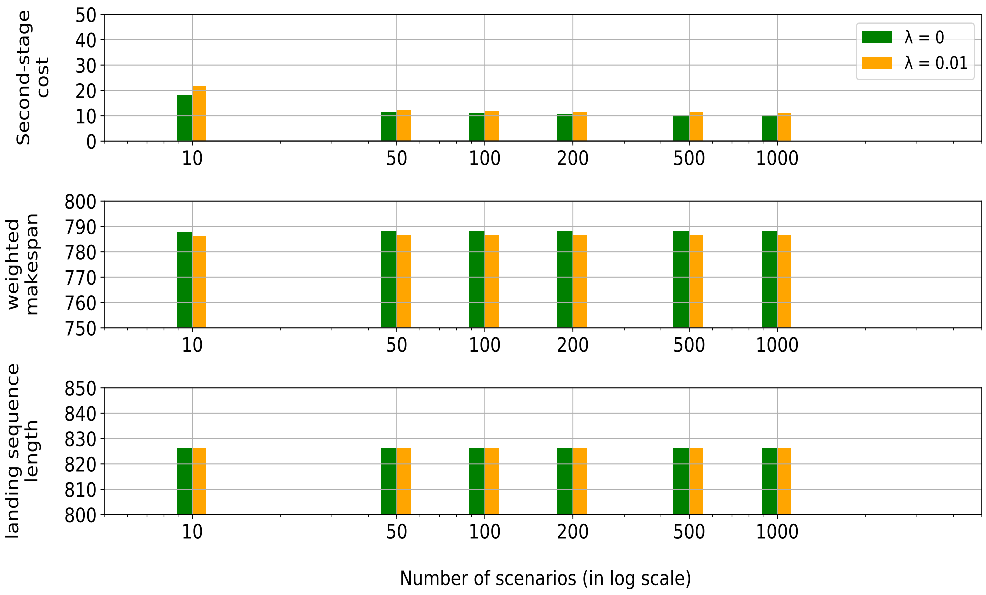


Fig. 6 Average landing sequence length, weighted makespan and second-stage cost for instance 1, narrow IAF time windows, small uncertainty and $\lambda = 0$ and 0.01.

small and large uncertainty, respectively).

On 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]. For each uncertainty amplitude, we test different numbers of replications within the

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)	38.8 (+ 80.5)	
	$v^{\star}$	832.4	835.8
	Validation	837.3	836.9
10	CPU (s)	75.7 (+ 171.1)	
	$v^{\star}$	829.4	832.2
	Validation	838.0	836.5
20	CPU (s)	165.6 (+ 397.9)	
	$v^{\star}$	828.7	832.2
	Validation	838.5	836.5
30	CPU (s)	246.9 (+ 615.7)	
	$v^{\star}$	828.7	832.2
	Validation	838.5	836.5

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

$n_{\mathcal{R}}$	solution	*min-Obj*	*min-Val*
5	CPU (s)	294.5 (+ 72.3)	
	$v^\star$	901.9	915.2
	Validation	919.5	917.9
7	CPU (s)	420.3 (+ 120.6)	
	$v^\star$	901.9	915.2
	Validation	919.5	917.9
10	CPU (s)	594.6 (+ 159.7)	
	$v^\star$	893.6	903.6
	Validation	919.5	917.0
20	CPU (s)	1190.0 (+ 321.8)	
	$v^\star$	889.7	903.6
	Validation	919.3	917.0

available time budget. Results of the replication-based approach for both small and large uncertainties 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 validation score. Figures 7 and 8 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 red.

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 (1000) within the time budget of 4 minutes. As a consequence, it outperforms both solutions from the replication-based approach. However, under large uncertainty, the scenario-based approach based on only 250 scenarios slightly exceeds the time budget of 10 minutes and performs worse than the *min-Val* solution.

As a conclusion, under small uncertainty, a fast solution method should be built on a scenario-based approach using a large number of scenarios (1000). However, under large uncertainty and lacking efficient deterministic solution method for the SAA problem, a replication-based approach may deliver better-quality solutions in a short time even with small numbers of scenarios (50) and replications (7) especially if the retained solution is selected following the validation-score criterion.

Fig. 7 Solution characteristics from replication-based approach in terms of the number of replications under small uncertainty

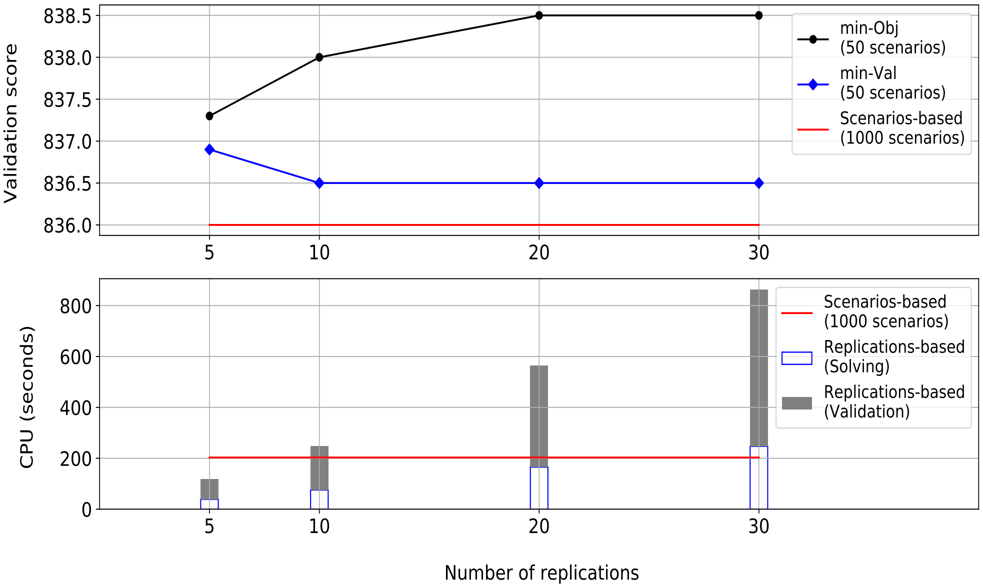
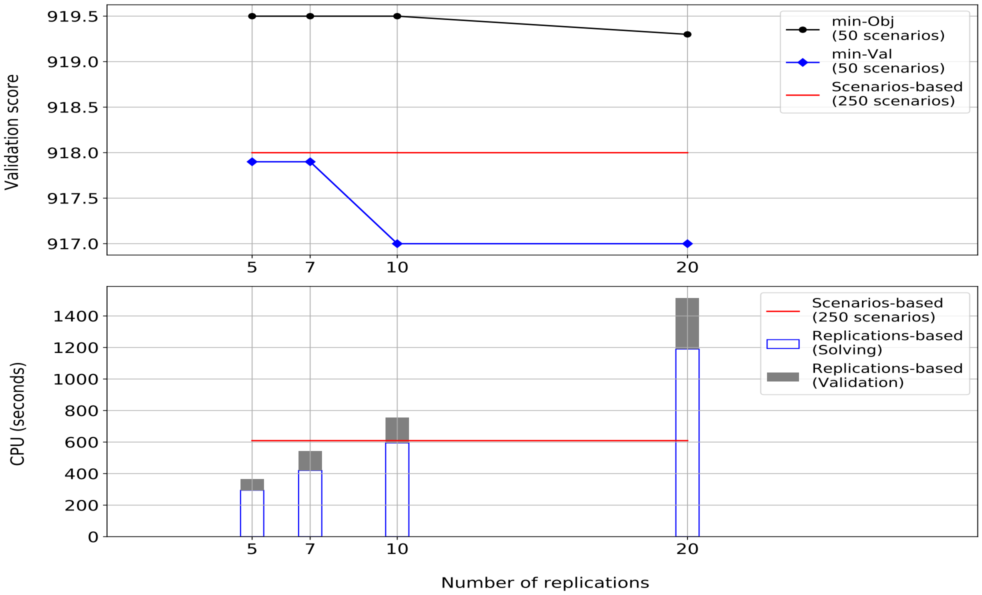


Fig. 8 Solution characteristics from replication-based approach in terms of the number of replications under large uncertainty



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). We solve a single SAA problem using 100 scenarios with 10 minutes as a time limit for CPLEX. 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, 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, $\lambda = 0$ and $n_{\mathcal{S}} = 100$

σ	Small	Large
CPU (s)	600.0 (+ 20.4)	600.1 (+ 20.9)
Status (Gap)	Feas. (11.5%)	Feas. (22.3%)
$v^\star$	1186.7	1351.9
Validation	1191.9	1358.6

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

In this subsection, we assume that ATCs in the terminal area sequence aircraft for landing in the same order in which they pass over the IAF. For each aircraft, the landing time is the same as the unconstrained landing time unless it violates the final approach separation with the leading aircraft (if any). In case of violation, the landing time is set to the earliest time that satisfies the separation with the leading aircraft. 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, i.e. scenario-based approach for small uncertainty and replication-based approach for large uncertainty, 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 two upstream situations: with and without optimization over the IAF. The first upstream situation consists in using our retained solutions (mainly the target times at the IAF) to schedule arrivals at the IAF. The second upstream situation, called the *baseline* situation, consists in scheduling the aircraft at the IAF according to their planned times at the IAF without any optimization. Hence, the target times at the IAF in the baseline situation are simply the planned times. For each situation, 10,000 scenarios are simulated and the following performance measures are computed:

- Average number of conflicts at the IAF, computed as the average number of separation violations over the IAF.
- Average ATC workload in the terminal area, computed similarly to the second-stage objective function as the sum of the absolute deviations of the target landing times with respect to the unconstrained landing times.
- Average landing rate per hour
- Average last landing time
- Average total delay in the terminal area, the delay for a single aircraft in the terminal area being computed as the (positive) deviation from its target landing time with respect to its unconstrained landing time.
- Average maximum delay in the terminal area

Values of these measures, for instance 1 with narrow time windows and $\lambda = 0$, are given in Tables 11 and 12 for the cases of small and large uncertainty.

Table 11 FCFS performance in the terminal area: instance 1, narrow IAF time windows, small uncertainty

σ	Small	
solution	Baseline	Optimized
Avg. IAF conflicts	4.3	0.3
Avg. ATC workload	787.4	10.3
Avg. landing rate	27	26
Avg. last landing	06:37:14	06:38:11
Avg. total delay	13 min 7 s	< 20 s
Avg. max. delay	3 min 55 s	< 10 s

Table 12 FCFS performance in the terminal area: instance 1, narrow IAF time windows, large uncertainty

σ	Large	
solution	Baseline	Optimized
Avg. IAF conflicts	4.0	1.5
Avg. ATC workload	745.4	93.9
Avg. landing rate	27	26
Avg. last landing	06:37:20	06:38:31
Avg. total delay	12 min 25 s	1 min 34 s
Avg. max. delay	3 min 43 s	1 min 0 s

Regardless of the uncertainty amplitude, our retained solutions clearly decrease the number of expected conflicts at the IAF, compared to the baseline situation where no change is made on the IAF planned times. Under small uncertainty, 93% less conflicts are expected at the IAF, and the expected ATC workload in the terminal is remarkably alleviated (by around 98%) for almost equal landing rates. Good, although smaller, decrease rates are also achieved under large uncertainty. On average, last landing times using our optimized solutions occur 1 minute later than without optimization. This may be due to the fact that the optimized schedule at the IAF is too sparse. Finally, we recall that a single tour in a holding stack is usually flown in 4 minutes. Measures of delay in the terminal area show that our solutions may avoid resorting to such holding stacks. Moreover, the average total delay experienced by the 10 aircraft in the terminal area become insignificant using our solutions. 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) to a linear holding applied when aircraft are still a few hours away from landing, as expected from an efficient E-AMAN.

V Conclusion and perspectives

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 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, narrow time windows at the IAF and 1000 second-stage scenarios gave good-quality solutions in a short solving time for small uncertainty (less than 4 minutes). Simulation-based validation experiments show that our retained solutions can decrease the number of expected conflicts over the IAF by more than 90%, and the expected ATC workload in the terminal area by 98% over a FCFS policy in the terminal area, while the landing rate remains nearly unchanged. Moreover, although our solutions do not enhance the expected landing rate, they significantly decrease expected aircraft delay near the terminal area. Some perspectives for future research include studying the mathematical formulation to find algorithmic enhancements that may decrease CPLEX solving times. To handle the dynamic case, an efficient solution method based on the rolling horizon algorithm may be proposed. Also, we plan to develop an efficient solution scheme to handle several IAFs and multiple runways. Finally, we may consider our instances with flight-dependent time windows over the IAF and flight-dependent uncertainty amplitudes, taking into account that, at the observation time (2 to 3 hours look-ahead time), some flights are already airborne while others are still on the ground.

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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[17] 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.

[18] 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.

[19] 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.