Università degli Studi di Milano-Bicocca
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dipartimento di Informatica Sistemistica e Comunicazione
Dottorato di Ricerca in Informatica - XXVI Ciclo

DETERMINISTIC AND STOCHASTIC OPTIMIZATION FOR HETEROGENEOUS DECISION PHASES IN THE AIR TRAFFIC DOMAIN

Luca Corolli
Ph.D. Thesis

Supervisor: Prof. Guglielmo Lulli
Tutor: Prof. Marco Antoniotti
Ph.D. Coordinator: Prof. Stefania Bandini

Anno Accademico 2012-2013

*Dedicated to those
who do not have access
to free education*

CONTENTS

ABSTRACT

Scheduling is a complex activity that is needed in a large number of fields, which can involve heterogeneous factors and that may have different goals. In this thesis, scheduling problems that involve the air traffic field at different phases are faced. At each phase, the different characteristics of the problems are considered, devoting special attention to uncertainty. Given the heterogeneous characteristics and goals of the problems, different models and methods are proposed to solve each of them.

The first analyzed phase is the strategic phase. It takes place around six months before the operation of flights, when they need to be assigned scheduled departure and arrival times at the airports where they operate. Future capacity realizations are very difficult to forecast at this phase, as capacity is influenced by weather conditions. A two-stage stochastic programming model with two alternative formulations is proposed to capture this uncertainty. Since the number of scenarios may be extreme, Sample Average Approximation is used to solve the model. The utilization of the proposed model allows to identify advantageous tradeoffs between schedule/request discrepancies, i.e., the distance between the allocated schedule and airline requests, and operational delays. This tradeoff can result in substantial reductions of the cost of delays for airlines. In the computational experiments, delays were reduced up to 45% on an instance representing a network of European airports.

The second considered phase is the tactical phase, which takes place on the day of operation of flights. At this time, complete flight plans need to be defined, specifying the route and operation times of flights. This is done considering two different sources of uncertainty. First, uncertainty on capacity availability is taken into account, similarly to the strategic phase. However, the number of capacity realization scenarios is now small, as they can be defined using available weather forecasts. The problem of minimizing delay costs considering this source of uncertainty is the Stochastic Air Traffic Flow Management problem. A two-stage stochastic programming model with two alternative formulations is proposed to solve this problem. An ad-hoc heuristic that takes advantage of the good structure of the model is used to solve problem instances within short computation times. The analysis of the Value of the Stochastic Solution shows that the proposed stochastic model can significantly reduce delay costs when bad weather affects the whole network. Second, the implicit uncertainty on the departure time of flights is taken into account. This kind of uncertainty involves operations that may cause a delay on the scheduled departure time of a flight. The flexibility of the scheduled time of departure, as well as the other flight operations, is determined defining time windows within which flights are granted capacity resources to operate. The narrower a time window, the more critical a flight operation. The problem of minimizing delay cost and maximizing time windows is faced by the Air Traffic Flow Management Problem with Time Windows. The problem is formulated with two alternative deterministic models, one of which is able to provide time windows in 40" on average for instances involving over 6,000 flights. Less conservative criteria to reserve capacity within time windows can also be used. Despite not granting the possibility for a flight to execute its operations at every instant of a time window, the implementation of these alternative criteria is shown to be viable. In fact, less than 0.14% of flights were subject to capacity shortages in the analyzed cases.

Finally, the operational phase takes place when operations are being executed. The goal at this phase is to manage the final departure times announced by flights – with uncertain information becoming deterministic – allowing them to depart at the announced time even if this time exceeds the assigned departure time window. This problem is named Real Time Flight Rescheduling with Time Windows. Resources are provided to flights that need them by reallocating previously reserved capacity with an algorithm that follows the Ration-By-Schedule mechanism. Both the practical usage of time windows and the impact of collaboration among airlines are studied. While airline collaboration limits time window flexibility up to some time before the scheduled departure of a flight, it can allow to reduce additional flight delays by over 14%.

This thesis is a first work that involves determining flight schedules from the moment of their definition to the time of execution of flights. Providing cost reductions by considering the different factors that influence each decision phase can lead to a global improvement of the management of flight operations, whose delays are very expensive in practice for airlines.

Chapter 1
INTRODUCTION

Decision making shapes both individuals and organizations. It ranges from simple everyday decisions to complex processes that involve a huge amount of data. This thesis focuses on a well studied category of decision making problems, i.e., scheduling problems. Decisions on schedules differ greatly depending on the time at which they are made. It is therefore possible to face problems more efficiently individually, by decomposing the whole decision making process into different decision phases, maintaining the coherence between decisions in a unique framework. Then, to solve each problem, different mathematical models and algorithms are developed. In particular, models explicitly consider the uncertainty that characterizes each decision phase. Decision making is subdivided into the three following phases.

Strategic phase
This is a long or medium-term decision making phase, and it is characterized by high uncertainty, i.e., the number of possible future scenarios is large. Decisions at this phase may involve actions of high economic importance. Some examples are planning the construction of new infrastructures, displacing long-term investments, creating or cancelling flight routes, etc. Uncertainty may characterize heterogeneous data, from the availability and cost of different resources, to the profit resulting from different investments. Since decisions involve future operations, it is in general possible to execute algorithms and models that are complex and require long computation times.

Tactical phase

This phase involves short-term problems. Decisions of high importance may have already been made at the strategic phase, however adjustments of previous decisions – such as the revision of existing schedules – based on more precise forecasts on uncertain data may critically improve system performance. Uncertainty is now characterized by a limited number of scenarios compared with the strategic phase. For example, if resource availability depends on weather conditions, uncertainty may even be completely removed based on accurate weather forecasts. Algorithms and models have limited time to be executed, as decisions involve operations that are scheduled to start in the near future. The specific amount of time available is application-dependent, and could be in the order of minutes.

Operational phase

This is the last decision making phase faced, as it takes place just before or during the execution of the operations. Uncertainty is now vanishing and deterministic information is being provided. Using this information, it may be possible to provide real-time adjustments to the execution of ongoing operations, for example by providing a better usage of available resources. Since these decisions are made while the operations are being executed, algorithms should provide results in real time.

From a modeling point of view, faced problems all belong to the general class of Resource-Constrained Project Scheduling Problems (RCPSPs). These problems can be focused on many different subjects by defining models with different characteristics and goals. In general, they involve the scheduling of projects, which are defined as sequences of jobs or operations, that need to use limited resources. The different specializations of RCPSPs may consider job preemptiveness, resource renewability, variability of resource availability, job multi-modality, multiple project scheduling, and different objectives. The interested reader may refer to the survey by Hartmann and Briskorn [1] for an overview on RCPSPs. More specifically, this thesis focuses on scheduling problems from the field of air traffic faced at each considered decision making phase. Specializing the RCPSP to this class of problems results in considering projects as flights and jobs as flight operations, with limited resources being sector and airport capacity.

This chapter unfolds as follows. In §1.1 the field of air transportation is analyzed, illustrating its current state and challenges. Related terminology is then introduced in §1.2. Air traffic problems belonging to each different decision making phase are discussed in §1.3. Some of these problems are modeled using stochastic programming, which is introduced in §1.4.

1.1 The air transport industry

The air transport industry is one of the biggest industries in the global economy. It involves all the activities needed for operating flights, both on the ground and in the air. Monetary and traffic figures show that it is a growing industry that is very challenging to deal with. EUROCONTROL's 2012 Performance Review Report [2] shows that the number of IFR (Instrument Flight Rules) flights in Europe almost doubled from 1990 to 2008, increasing from approximately five million flights to over ten million flights per year. This increasing trend showed a two-year decrease period between 2001 and 2002, as a result of the September 11 terrorist attacks in the United States. After 2008, the economic crisis hit the air traffic industry, resulting in a drastic decrease of the number of operated flights in 2009. Operated flights increased again in 2010 and 2011, then decreased in 2012, showing the current instability of the European air transport industry. However, the 2013 STATFOR seven-year forecast [3] expects European air traffic to reach pre-economic crisis levels by 2016, and a continuation of the growth afterwards. European air traffic is expected to double by 2035 in the most likely scenario, with 12% of traffic not accommodated. Also, the top 20 airports are expected to become heavily congested, handling over 150,000 departures per year, a level of traffic achieved in 2013 only at eight airports in Europe [4].

Many stakeholders generate revenue in this industry, mainly airports, airlines and aircraft manufacturers. In 2011, 5.4 billion passengers were served by the 1,345 Airports Council International (ACI) member airports worldwide [5]. This figure corresponds to an increase of 8% over 2010. According to the ACI Economics Survey 2011 [6], worldwide total airport income in financial year 2011 reached $101.8 billion, an increase of 7% on the previous financial year. Airport operating expenses amounted to 55% of the total revenue. Airlines, on the other hand, enjoyed industry-wide net profits of around $8 billion in 2011 at a global level. These profits are only half of those recorded in the previous year, but still represent a reasonable outcome when compared against recent historical results. The core reason for the dip in net profits in 2011 is that the rise in costs (10.8%) outstripped that of revenues (9.4%) compared with the previous year. High fuel costs were the main contributory factor accounting for 30% of total costs in 2011. Non-fuel expenses also rose to their highest level in the last seven years to $405 billion [5]. According to the International Air Transport Association (IATA), its member airlines collectively recorded an air cargo decline – measured in Freight Tonne Kilometres (FTKs) – of nearly 1% in 2011 over 2010 levels. This was mainly due to a slowdown in export demand from the Asia-Pacific region to its major consumption markets in Europe and North America [5]. Finally, figures related to aircraft manufacturers report a turnover of the European aeronautic sector in 2010 equal to 106.6 B€, an increase of 6.2% over 2009. Despite this increase, the number of persons employed in aeronautics decreased by 2% over 2009, reaching 458,700 units.

A lot of different issues are faced in the field of air transportation. An obvious issue for all stakeholders is that of generating revenue. This is an issue strictly related with all other air transportation issues and operations. Effectively facing environmental issues, for example, can allow to reduce both air pollution and fuel consumption. A reduction in fuel costs can provide great savings to airlines, since they amounted to 30% of total airline costs in 2011. Safety and security of flights are also crucial for the whole market. Disastrous events can in fact heavily influence the perception of air transportation, as proved by the worldwide decrease of air traffic figures that followed the September 11 attacks. Another key issue in air transportation is delay minimization. Delays in the air traffic system are not only expensive in terms of time, but also in terms of money. Air Navigation Service (ANS) costs are divided into two categories: ANS provision costs, which derive from terminal and en-route charges, and ANS quality of service related costs, that derive from flight inefficiencies and from Air Traffic Flow Management (ATFM) delays. ATFM regulates air traffic in order to avoid exceeding available capacity in handling traffic, using capacity efficiently. In 2012, estimated ANS provision costs in Europe amounted to a total of 8.3B€, while ANS quality of service related costs amounted to a total of 4.6B€, with 900M€ resulting from ATFM delays [2]. The University of Westminster estimates ATFM delays in Europe to cost 83€ per minute [7]. The problems faced in this thesis are all related to the minimization of air traffic delays, using some novel approaches based on deterministic and stochastic optimization that focus on different sources of uncertainty. This approach is consistent with the Single European Sky ATM Research (SESAR) Work Package E (WP-E) theme 4 (Information Management, Uncertainty and Optimization) research project [8]. Next, definitions of terms commonly used in air traffic problems are introduced.

1.2 Air traffic definitions

The first air traffic definitions described in this section are those involving the resources needed for the execution of flights. Delays in air traffic are caused by imbalances between the demand and the availability of needed resources. These imbalances result in congestion, which typically arises at airports or in the airspace around them. Airports are particularly critical as the number of runways is limited, and it may not be possible to expand them or to build new ones due to physical limitations. This results in the definition of a maximum number of departures, arrivals, or a combination of the two at any time at an airport. Each of these numbers defines the departure, arrival, and total *airport capacity*. In the long term, planning the increase of current airport capacity is an absolute prerequisite to cope with the projected increase in traffic demand. EUROCONTROL reports that capacity provided in 2011 was still at the level provided in 2007, thus indicating insufficient attention to this problem and motivating the pessimistic congestion forecasts at the top European airports. However, at least in the short-medium term, optimizing the use of existing capacity is necessary in order to alleviate congestion.

The airspace is subdivided into *air sectors* – or simply *sectors* – that are managed by air traffic controllers. Congestion in the airspace results from two main factors. First, from the physical limitations of the air sectors in which safety rules for flying must be respected at all times. Second, from the number of available air traffic controllers who can monitor an air sector at a specific time. Physical limitations of the airspace are of particular relevance around airports, as that is where flights converge in order to depart or land. Similarly to airports, the maximum number of flights that may simultaneously be managed in an air sector defines *sector capacity*.

The air traffic system has very high internal interdependencies, i.e., delay on a flight may cause disruptions in the whole system. In fact, each flight is *connected* to a set of other flights at its destination airport. This *flight connectivity* results from the resources that are shared between flights, e.g., the aircraft, the flight crew, etc. Practically, flight connectivity results in the need for a connected flight to wait for one or more incoming flights to land and provide it with the complete set of resources it needs to depart. The amount of time separating each pair of connected flights is called *turnaround time*.

To efficiently manage flight operations, different control options may be considered. In particular, three fundamental control options are now discussed. The first is called *rerouting*, and it is the possibility to deviate a flight from its "standard" *route* to a different one. A route is defined as the trajectory that a flight follows to go from origin to destination. The trajectory can be defined in different ways depending on the context of application. For modeling purposes, it is generally defined as the ordered set of sectors crossed by a flight to fly from origin to destination. Rerouting can be a useful control option to avoid congested air sectors. The second fundamental control option is *ground holding*. This control option involves moving delay from the airspace to the ground, whenever possible. Ground holding is useful as delay that takes place in the air, called *airborne delay*, is more expensive than delay that takes place on the ground, which is called *ground holding delay*, or simply *ground delay*. In fact, when an aircraft is waiting to depart on the ground, it does not consume any fuel, which is expensive and would have been consumed in the air if the delay took place after the departure. Also, a flight waiting to depart at its airport of origin does not congest the airspace and does not need to be supervised by any air traffic controller for its safety. The third control option considered is *speed control*. With this option, the speed of an aircraft can be changed along its route, resulting in a variable sector crossing time. This option can allow to move flight delays in space, to avoid causing further congestion to already congested sectors. For details on additional available control options, the interested reader may refer to Ball *et al.* [9]. In the following section, air traffic problems at each considered decision making phase are discussed.

1.3 Decision making at different phases for air traffic problems

The different air traffic problems faced at each decision making phase are now illustrated. In particular, the different objectives and types of uncertainty considered are highlighted.

Strategic decision making

The strategic phase is the first decision phase considered, and it takes place around six months before the execution of flights. It is discussed in chapter 2. At this phase, timetables at different airports are determined. Airlines formulate requests to operate flight departures and arrivals at specific times, and an initial flight schedule is determined taking the uncertainty on capacity availability into account. Considering uncertainty on capacity availability is very important, as at this phase the number of possible scenarios is – in general – very high, and it could be in the order of billions. Since the main source of uncertainty on capacity availability is weather conditions, it is only possible to use historical data or seasonal forecasts – which are very unprecise – to define the scenarios. Furthermore, since the proposed schedule is an initial one and it will be subject to further revision, and considering that capacity availability is mostly critical at airports, only airport capacity is considered. This way, it is possible to provide an initial flight schedule at considered airports in a simplified way. Computation times allowed for problems at this phase may be in the order of days.

Tactical decision making

The second decision phase considered is the tactical phase, which is discussed in chapter 3. This phase takes place a few hours before the execution of flights. Two different problems are faced at this phase. First, the problem of providing precise and robust flight plans is faced, considering the whole set of airport and sector capacity resources needed for the execution of flights. Uncertainty on capacity availability is considered again. However, at this phase, it is generally possible to formulate forecasts on capacity availability that are more precise, limiting the number of scenarios by deriving them from available weather forecasts. Second, the problem of providing flexibility to flight operations is faced. This flexibility involves another type of uncertainty, which is the implicit uncertainty on the departure time of flights. In this case, no stochastic information is explicitly considered. The goal is to define time windows within which capacity to execute flight operations is granted. This allows for a flexible execution of the operations that does not involve the reassignment of capacity resources. In fact, the delayed departure of a flight usually corresponds to the need for its operating airline to request the assignment of new capacity resources to depart. Time windows allow to avoid many of these requests, making the system more flexible and highlighting which flights are critical to operate. Computation times for both problems should not exceed 15-30 minutes.

Operational decision making

The third and final decision phase considered is the operational phase, which is discussed in chapter 4. Decisions are made just before the departure of some flights, while other flights are already being executed. Uncertainty on capacity availability is not considered, as at the time of operation of flights it can be considered deterministic. However, uncertainty on the departure time of flights may still be present. In fact, problems arising shortly before the planned departure of a flight, typically related to airport operations, may force airlines to delay departures. In this context, time windows defined at the tactical phase are considered as a starting point: departures can take place at any time within the assigned time windows. Should a departure time window not be met, the corresponding flight needs to be assigned new capacity to depart. The goal of the considered problem is to avoid wasting capacity resources that are not used by other flights, reassigning them to those flights that need to use them due to exceeding their own time windows. These reassignments should be made in real time, so algorithms should be designed to provide solutions in a matter of seconds.

Different types of models and algorithms are used at each decision phase. Both deterministic and stochastic optimization models are used, depending on the requirements and the type of uncertainty considered by each problem. For the sake of completeness, stochastic optimization is introduced in the next section.

1.4 Introduction to stochastic programming

When uncertainty is an important factor and constraints on computational times are not too strict, problems can be modeled using stochastic programming, which can provide better solutions over deterministic programs. For a broad overview of the main themes and methods of stochastic programming, the interested reader may refer to the introductory textbook by Birge and Louveaux [10]. In this thesis, two-stage stochastic integer programming (SIP) models are used to model problems involving uncertainty on airport or sector capacity. In this type of models, a decision must be made "here and now", in the first-stage of decision making, when uncertainty is still present. This uncertainty is modeled using a probability distribution of the random variables. When uncertainty is resolved, in the second-stage of decision making, a recourse action is taken. The general two-stage SIP model can be formulated as follows:

$$\begin{aligned} z = \text{Min } & c^\top x + \mathbb{E}[f(x, \tilde{\omega})] \\ \text{s.t. } & Ax \geq b \\ & x \in X. \end{aligned} \tag{1.1}$$

In formulation (1.1), decisions are made on the first-stage decision variables vector $x \in X \subseteq \Re_+^{N_1}$. The objective function is also characterized by the first-stage cost vector $c \in \Re^{N_1}$ and the recourse function $f(x, \tilde{\omega})$, with $\tilde{\omega}$ being a multivariate random variable defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. The underlying probability distribution of $\tilde{\omega}$ is assumed as discrete with a finite number of realizations (scenarios) ω and corresponding probabilities $p(\omega)$, $\omega \in \Omega$, with Ω being the set of all scenarios. Furthermore, first-stage constraints are characterized by the first-stage constraint matrix $A \in \Re^{M_1 \times N_1}$, the first-stage right-hand side vector $b \in \Re^{M_1}$, and the set X which defines integer or binary restrictions on some components of x.

For each scenario $\omega \in \Omega$, the recourse function $f(x, \omega)$ is given by the following second-stage mixed integer program (MIP):

$$\begin{aligned} f(x,\omega) = \text{Min } & q(\omega)^\top y(\omega) \\ \text{s.t. } & W(\omega)y(\omega) \geq h(\omega) - T(\omega)x \\ & y(\omega) \in Y. \end{aligned} \tag{1.2}$$

In formulation (1.2), decisions are made on the second-stage decision variables vector $y(\omega) \in Y \subseteq \Re^{N_2}$. In the objective function, $q(\omega) \in \Re^{N_2}$ is the recourse cost vector. The constraints are also characterized by the recourse matrix $W(\omega) \in \Re^{M_2 \times N_2}$, the second-stage right-hand side vector $h(\omega) \in \Re^{M_2}$, and the technology matrix $T(\omega) \in \Re^{M_2 \times N_1}$. When the recourse matrix is independent of the scenario realization, the recourse function is said to be fixed recourse. This is the case for all the models considered in this work, as the only parameter to change with the scenario realization is the second-stage right-hand side vector. Finally, set Y defines integer or binary restrictions on some components of $y(\omega)$.

It is always possible to solve a stochastic programming model through its Deterministic Equivalent Problem (DEP). The DEP of a problem is a single deterministic formulation that is equivalent to the two-stage formulation from (1.1) and (1.2), and it can be solved using any optimization solver. The DEP has the following formulation:

$$\begin{aligned} \text{Min } & c^\top x + \sum_{\omega \in \Omega} p(\omega) q(\omega)^\top y(\omega) \\ \text{s.t. } & Ax \geq b \\ & W(\omega)y(\omega) \geq h(\omega) - T(\omega)x \quad && \forall \omega \in \Omega \\ & x \in X \\ & y(\omega) \in Y && \forall \omega \in \Omega \end{aligned} \tag{1.3}$$

Solving a problem through its DEP, however, may be inefficient. The number of second-stage variables in the DEP is equal to the sum of the number of second-stage variables over all scenarios. The same applies to second-stage constraints. When the number of scenarios is very large, the DEP may become too large to be solved in reasonable computation times. Decomposition approaches or ad-hoc algorithms that take advantage of a specific structure of the mathematical formulation may improve the computation times provided by solving the DEP of a problem. In this thesis, an ad-hoc heuristic is proposed to solve instances of the Stochastic Air Traffic Flow Management problem, taking advantage of their special structure, see §3.3. Decomposition approaches for multi-stage stochastic programming are generally classified as either stage-wise or scenario-wise. The stage-wise approach is usually based on Benders decomposition [11] or L-shaped decomposition [12]. In this type of decomposition approach the problem is decomposed by decision stage, starting from the first-stage and then evaluating the effect of such decisions in the second-stage. This approach can be extended to an arbitrary number of decision stages. In the works presented in this thesis, only two-stage models are discussed. In scenario-wise decomposition, first-stage decision variables are duplicated to create several independent problems. Nonanticipativity constraints are then enforced to ensure that the first-stage decision is the same for all scenarios. Such constraints are relaxed using Lagrangian relaxation and a solution to the Lagrangian dual is then searched with specific algorithms.

A different computational problem arises when the number of possible scenarios is too high to evaluate first-stage decisions for all of them. This is the case for many strategic problems. A possible approach for solving such problems involves the use of sampling techniques, which consider only randomly selected subsets of the set of scenarios in order to obtain approximate solutions. Sampling may be *interior* or *exterior*. Interior sampling is performed within the algorithm whenever needed, using a different sample each time. Examples of such algorithms are provided by the L-Shaped method with embedded sampling proposed by Infanger and Dantzig [13], and the stochastic decomposition method by Higle and Sen [14, 15], to mention a few. In exterior sampling algorithms, on the other hand, a sample is defined from the set of scenarios and a corresponding approximation of the recourse function is determined. The approximate objective is called *sample average approximation* (SAA) of the recourse function, and optimization is performed without performing further sampling. SAA is used in this work to solve instances of the Time Slot Allocation Problem under Uncertainty, and is explained into detail in §2.3.

When time requirements are stricter, it may not be possible to use stochastic programming. However, it may still be possible to take uncertainty into account, formulating solutions based on deterministic data that are designed to do so. In this thesis, time windows are used to grant flexibility to flight operations. A time window is a set of contiguous time instants within which it is possible to execute operations. The initial instant of a time window is the scheduled time of operation, while the remaining instants of the time window provide flexibility to operations. Time windows are constructed by considering deterministic data, e.g., initial flight plans and expected capacity availability. Their width, however, allows to manage uncertain delays flexibly. Since they are determined using a deterministic model, they are compatible with short-term decision making.

In the following chapters, all discussed problems are faced, subdivided by the decision phase they are faced at. Different models and algorithms to solve them are proposed, and related computational results are presented.

Chapter 2
STRATEGIC DECISION MAKING

In this chapter, a strategic problem from the field of air traffic, the Time Slot Allocation Problem Under Uncertainty (TSAPU) is discussed. First, the problem and related literature are described in §2.1. Then, in §2.2, a stochastic programming model to solve the TSAPU is presented, providing two alternative formulations. The model is solved on realistic instances using Sample Average Approximation (SAA). This method is illustrated in §2.3, while the computational results from tests performed on a set of instances representing European airport networks of different sizes are discussed in §2.4. Finally, in §2.5, the conclusions and future directions of work on the TSAPU are summarized.

2.1 The Time Slot Allocation Problem Under Uncertainty

Congestion in air traffic causes very high expenses for airlines and aircraft operators every day. It usually arises at airports or in the airspace around them. Airspace congestion is more relevant in Europe than in the United States. Despite being *fully coordinated*, the most congested airports in Western and Central Europe suffer persistent congestion. Full coordination of an airport means, essentially, that the number of flights scheduled at the airport per hour (or other unit of time) is not allowed to exceed the *declared capacity* of the airport [16]. Here, a coordinator is appointed to allocate slots to airlines and other aircraft operators using or planning to use the airport as a means of managing available capacity. In the long term, planning the increase of current airport capacity is an absolute prerequisite to cope with the projected traffic demand. However, at least in the short-medium term, optimizing the use of existing capacity is necessary in order to alleviate congestion. Properly managing available airport capacity needs to start at the strategic phase, when the initial flight schedules are defined under high capacity availability uncertainty.

Several approaches have been proposed to alleviate congestion and resolve demand-capacity imbalances. Strategic approaches, such as the use of slot auctions or congestion pricing, are mostly administrative or economic in nature, and try to alleviate congestion by modifying spatial or temporal traffic patterns, see Brueckner [17], Fan [18] and Raffarin [19] among others. The work presented in this chapter focuses on strategic initiatives aimed at allocating slots among airlines and aircraft operators efficiently. A *slot* is defined as the amount of capacity needed by an aircraft to land or take-off. Its assignment to a flight corresponds to the permission given by a coordinator for a planned operation to use the full range of airport infrastructure necessary to arrive or depart at a fully coordinated airport on a specific date and time [20].

Slot allocation is *de-facto* handled according to the IATA guidelines. Since 1974, IATA has provided the global air transport community with a single set of standards for the management of airport slots, outlining policies, principles and processes for slot allocation. These guidelines are the result of consultation between airlines and airport coordinators, reflecting the proven best practice for the coordination and management of airport slots [20]. For the allocation of slots at European Union airports, IATA guidelines are enriched with further conditions that aim to encourage the efficient use of airport capacity through the optimal allocation of slots [21].

European Commission regulation 95/1993 guarantees that slot allocation is based on neutral, transparent and nondiscriminatory rules. In this regulation process, each Member State has the duty of appointing a schedule coordinator at airport or national level with the ultimate mission to operationalize, coordinate, supervise, and arbitrate the slot allocation process. The fundamental principle of the slot allocation process is the *grandfather right*, i.e., the right of an airline to keep a slot from the preceding equivalent season. This right is granted if and only if such a slot was used at least 80% of the time (*use-it-or-lose-it* rule). However, this procedure is far from being efficient. Indeed, as reported by ACI Europe, unsatisfied/unaccommodated demand, overbidding, late return of unwanted slots, flights operated significantly and repeatedly off slot time (*off slot*), and failure to operate allocated slots (defined as *no shows*), are all factors pointing or contributing to the inefficient allocation and use of an already insufficient resource, see Zografos *et al.* [22]. Several amendments to the EC regulation have been adopted or proposed to ensure the fullest and most flexible use of limited capacity at congested airports. For instance, regulation (EC) 793/2004 was adopted to strengthen the coordinator's role and the monitoring of compliance, i.e., the usage of slots with respect to the allocation, to verify that airlines do not use slots in a significantly different way than allocated by the coordinator.

Based on the UK experience, the Commission is also considering to change the current regulation. This is aimed at introducing market-based mechanisms across the EU, provided that safeguards to ensure transparency and undistorted competition are established, including greater independence for slot coordinators. Several studies that analyze the introduction of market-based mechanisms in the slot allocation process are present in literature. Rassenti *et al.* [23] developed a sealed-bid combinatorial auction for the allocation of time slots to competing airlines. A similar approach was proposed by Ball *et al.* [24]. Kleit and Kobayashi [25] and Fukui [26] examined whether slot markets have resulted in anticompetitive activities with restricted market entry and service expansion by other carriers, especially rival carriers. Verhoef [27] looked into some alternative economic instruments for managing congestion at airports, notably slot sales and slot trading. In the last few years, secondary trading received a lot of attention by the research community. Some of these studies were commissioned by the EU Commission as a guidance on the possible market and legal impacts of the introduction of such trading. For instance, the Mott MacDonald Group [28] analyzed into detail the likely effects of the introduction of secondary slot trading into Community legislation, while Starkie [29] examined the arguments for and against a secondary market in slots, focusing on evidence from U.S. airports. Furthermore, Pellegrini *et al.* [30] proposed a market mechanism for secondary trading based on budget balanced combinatorial exchange.

In this chapter, the focus is on the judicious allocation of time slots to aircraft operators with the purpose of obtaining "effective" and "reliable" airline schedules. Schedules are "effective" because slots are allocated according to airline preferences. Currently, slot allocations at different airports are independent. However, coherently allocating the departure and arrival slots at origin and destination for each flight is a widely recognized issue. Furthermore, it is necessary to consider interdependencies among flights operated by the same airline. The approach herein proposed, based on stochastic programming, allocates slots at all airports simultaneously, considering a true network of airports thus guaranteeing the coherence of the final result, i.e., the flight schedule. The difference between airline requests and the schedule, i.e., the allocated time slots, is referred to in this work as “schedule/request discrepancies”. Also, schedules are “reliable" because the proposed model takes the inherent uncertainty affecting airport capacity into account.

The slot allocation problem is currently solved by slot coordinators at each single coordinated airport independently, using a First-In-First-Served discipline. Issues related to airport network dependencies and slot complementarity are not immediately faced by slot coordinators, as they are resolved *off-line* at the IATA Worldwide Scheduling Conference, which is held biannually. Few approaches have been proposed for this specific problem. For example, Kösters developed a heuristic procedure [31] that addresses the interrelationship between slot demand and the time difference between assigned and requested slots in order to predict the latter depending on slot utilization. Another relevant work was developed by Zografos *et al.* [22]. The authors proposed a model for the single airport slot allocation problem, implementing existing EU/IATA scheduling rules and coordination procedures. They applied their work to real data from three different Greek airports, showing that it is possible to reduce the total amount of schedule/request discrepancies up to over 95%, compared to that assigned in practice without using optimization models. This is a very important result, as it shows that mathematical methods can greatly benefit the time slot allocation process. Finally, Pellegrini *et al.* [32] compared different metaheuristic algorithms to solve the slot allocation problem.

A stochastic programming model for the time slot allocation problem with two alternative formulations is proposed in this chapter. This model extends the deterministic single airport model by Zografos *et al.* [22] in the following directions. First, the new model simultaneously allocates time slots on multiple airports explicitly considering – in addition to other operational constraints – the coherence between the departure time slot at the airport of origin and the arrival time slot at the airport of destination of each flight. Second, it takes stochastic capacity availability into account. This is a relevant issue since time slot allocation is performed at a strategic phase, when the uncertainty on available capacity is extremely high, i.e., the number of possible capacity availability scenarios is extreme. The proposed stochastic optimization approach provides robust solutions to the problem by balancing the immediate costs generated by schedule/request discrepancies with future costs of delays that one can expect to be assigned using a proposed flight schedule on the days of operation.

Finally, it is important to note that, as a by-product, the proposed model is able to address an issue of great concern for slot coordinators. Indeed, slot coordinators are called for fixing the *declared* capacity of fully coordinated airports, i.e., the capacity that is available at each airport to accommodate the demand. Such values of capacity are currently assessed by a thorough demand and capacity analysis whenever there are significant changes in airport infrastructure, operational policies, or demand patterns. The problem of defining airport capacity is also faced in literature. Churchill *et al.* [33] proposed a stochastic optimization model with three alternative formulations to determine the number of slots that should exist at an airport during different time intervals across the course of a single day. Their model explicitly accounts for differing slot valuations and hedges against the variety of capacity outcomes that are likely to occur, providing responsiveness to demand and robustness against uncertain outcomes. Slack is built into the profiles where it is clear that a high probability of congestion later in the day could suffer greatly from residual queues from earlier in the day. Zografos *et al.* [34] focused on runway complexes explicitly taking the stochastic nature of aircraft operations into account. The authors proposed a fast and easy-to-use analytical model that estimates the capacity of a wide range of runway complexes, capturing all the major parameters that affect it. Furthermore, the model provides an estimate of the potential capacity benefits from sequencing arriving aircraft. Andreatta *et al.* [35] proposed two macroscopic decision support systems for airport strategic planning on airside and landside, discussing their connectivity. The airside decision support system, called MACAD, uses the model developed by Zografos *et al.* [34] as an internal module. The landside decision support system is called Simple Landside Aggregate Model (SLAM) and is an analytical aggregate model for estimating capacity and delays in airport passenger terminals, see Brunetta *et al.* [36]. Andreatta *et al.* [37] discussed the implementation of SLAM at the Athens International Airport, as well the implementation of an enhanced version of SLAM, see Brunetta *et al.* [38]. The airside and landside systems provide estimates of the capacity and performance of the airport quickly and with little effort.

Airport capacity values can be obtained from the solutions provided by the model presented in this chapter. In fact, the total number of departures and arrivals at airports, served in periods of demand peaks, can be interpreted as values of declared capacity, which is bounded by the nominal capacity. In this case, the declared capacity of an airport will not be a static value, but it will be time-dependent, i.e., it may change period by period on each day of operation thus guaranteeing the fullest and most flexible usage of limited capacity at congested airports. The new two-stage stochastic programming model for the TSAPU is described in the following section.

2.2 Stochastic optimization model

In this section a stochastic programming model with two alternative formulations to solve the TSAPU is presented. Given the nominal capacity of each airport, i.e., the number of available time slots, and airline requests for time slots in order to operate specific flight movements, the TSAPU allocates time slots to airlines and aircraft operators. This results in a flight schedule that allocates time slots with the goal of accommodating airline preferences, i.e., slot requests, by minimizing the difference between the allocated and the requested slot times, i.e., the schedule/request discrepancies. However, on each day of operation, available capacity can be much less than the nominal one, e.g., as a consequence of bad weather conditions. In this case, large delays may have to be assigned to flights in order to resolve demand-capacity imbalances. Therefore, a tradeoff exists between the schedule/request discrepancies and expected delays to be assigned on the day of operation. A very tight schedule operating at nominal capacity is likely to be subject to delays on the day of operation when capacity is reduced. The TSAPU aims at finding a schedule that results in a good compromise for the described tradeoff.

One of the fundamental aspects of the proposed model is the fact that it considers a network of interconnected airports, taking downstream effects of local decisions into account. Since a movement is either the departure or the arrival of a flight taking place at an airport, a pair of movements (m_1, m_2) is associated with each flight, where m_1 is the departure movement from the airport of origin and m_2 is the arrival movement of the same flight at the destination airport. The elapse of time between the departure time slot at the airport of origin and the arrival time slot at destination corresponds to the flight time. This amount of time is fixed, as it corresponds to the scheduled duration of the flight. In view of this fact, allocating a specific time slot for the execution of the departure of a flight explicitly fixes the allocation of an arrival time slot at the airport of destination of the flight. Decisions are therefore not merely local, as they have effects on the whole network. Furthermore, each movement is requested on a set of days, to ensure that the same time slot is assigned on each day of operation to the same movement. Therefore, assigning a time slot to a movement affects multiple days simultaneously. From a modeling point of view, this helps to keep the model's dimensions at a reasonable size, as the number of decisions to be made is the same for a movement requested on a single day or on a large set of days.

Flight connectivity, that makes sure that the turnaround time for each pair of connected flights is respected, is also considered by the TSAPU to define schedules. Furthermore, it is important to highlight that the proposed model is flexible enough to accommodate capacity constraints deriving from operational considerations. Despite decisions related to the optimal mix of departure/arrival operations being of daily/operational nature, they have a remarkable impact in setting the declared capacity of an airport. By including these “operational constraints", the declared capacity of an airport will not be a static value but rather time-dependent, i.e., it may change period by period on each day of operation thus guaranteeing the fullest and most flexible usage of limited capacity at congested airports. Finally, the TSAPU explicitly models uncertainty, which affects – predominantly – airport capacity. In the following subsection, an example that highlights the benefit of incorporating uncertainty within the mathematical model is illustrated.

2.2.1 Example

An example of the functioning of the proposed model is now illustrated, to highlight how considering uncertainty on capacity availability can lead to decrease flight delays. Consider an airport with capacity constraints defined at each hour, with a limit of two departure movements, two arrival movements and three total movements per hour. Suppose that different airlines formulate the following movement requests at this airport: (a) arrival at 8:30, (b) departure at 9:00, (c) arrival at 9:20, (d) departure at 9:50, (e) arrival at 10:20, (f) departure at 10:30. These requests correspond to the aggregated hour by hour demand reported in Table 2.1. It is easily possible to verify that this demand respects the airport's capacity constraints, therefore in a deterministic setting the optimal solution allocates all requested time slots to airlines .

Now consider uncertainty on available capacity by defining two equiprobable scenarios. In the first scenario, which has probability 0.5, nominal capacity is respected. In the other scenario, capacity is decreased by 33% with probability 0.5. Since capacity assumes integer values, this corresponds to having departure and arrival capacity constraints both to one flight movement per hour, and the total capacity to two flight movements per hour. Considering this information and 15' wide time slots, the delay on the day of operation

Table 2.1: EXAMPLE: MOVEMENT DEMAND

Time horizon	**Arrivals**	**Departures**	**Total**
8-9	1	0	1
9-10	1	2	3
10-11	1	1	2

of flights – when flights can only be delayed and cannot depart ahead of schedule – using the schedule defined only considering deterministic information is:

- With nominal capacity: 0
- With reduced capacity: movement d is delayed to the 10:00 time slot (1 time instant) and movement f is delayed to the 11:00 time slot (2 time instants), resulting in a total of 3 instants of delay

The average delay is therefore equal to $0 \cdot 0.5 + 3 \cdot 0.5 = 1.5$ time instants, i.e., 22'30". This delay can be reduced by considering capacity scenarios when defining the schedule. For example, it is possible to define a schedule that is feasible for all capacity scenarios, i.e., providing no operational delay, by simply moving departure movement b from the 9:00 to the 8:45 time slot. This results in a discrepancy between the schedule and airline requests of one time instant (15'), which is less than the average operational delay of 1.5 time instants (22'30") on the day of operation of flights obtained by using the schedule formulated by only considering deterministic information. From this small example it is also possible to see the tradeoff between the schedule/request discrepancies (0' considering deterministic information *versus* 15' considering stochastic information) and the delay on the day of operation (22'30" considering deterministic information *versus* 0' considering stochastic information).

2.2.2 Formulations

The two formulations of the new two-stage stochastic programming model for the TSAPU are now illustrated. The two formulations are the same in the first-stage, where flights are scheduled according to airline requests, satisfying nominal capacity at all airports. The objective is to minimize the schedule/request discrepancies, plus the estimate of future delays that will arise by choosing a specific schedule. This estimate is provided by the second-stage recourse function, for which the two formulations differ. Their goal is to provide an estimate of the delay on the day of operation of flights. The first formulation is called *simplified recourse*. It does not consider the downstream effect of delays over time, providing a rounding down estimate of future delays. The other formulation is called *time-linked recourse*, and it takes delay propagation between consecutive time instants into account. The simplified formulation has the advantage of being more computationally efficient than an exact approach, as it is a mere count of instant by instant delays, and therefore separable into many different small subproblems. The time-linked formulation, on the other hand, provides a higher level of accuracy. However, this comes at the cost of having longer computational times to solve instances of the problem. This is due to the fact that the time instants are not independent of one another as in the simplified recourse formulation, but are linked, thus making the problem inseparable by time. In the following, the first-stage subproblem – which is common to both formulations – and the recourse subproblem for each formulation are illustrated.

Formulation of the first-stage

To formulate the problem, the time horizon is assumed as fixed and subdivided into equal size contiguous time slots. This assumption is typical of nearly all Air Traffic Management (ATM) formulations and agrees with practice. The notation used in both formulations of the TSAPU model is now introduced.

Notation

$\mathcal{T}$: Set of time slots, indexed by t

$\mathcal{D}$: Set of scheduled days, indexed by d

$\mathcal{A}$: Set of airports, indexed by a

D_a^d : Set of departure movements taking place at airport $a \in \mathcal{A}$ on day $d \in \mathcal{D}$

A_a^d : Set of arrival movements taking place at airport $a \in \mathcal{A}$ on day $d \in \mathcal{D}$

$\mathcal{M}$: Set of all movements ($\mathcal{M} = \cup_{a \in \mathcal{A}, d \in \mathcal{D}} D_a^d \cup A_a^d$), indexed by m

DC_a, AC_a, TC_a : Set of departure, arrival and total capacity constraints for airport $a \in \mathcal{A}$, indexed by c, each spanning over t_c time slots

$\mathcal{F}$: Set of movement pairs (m_1, m_2) corresponding to the operation of a specific flight. The flight time $t_{m_{12}}$ is the time difference between the two movements

$\mathcal{P}$: Set of movement pairs (m_1, m_2) corresponding to consecutive flights, where $m_1 \in \mathcal{M}$ is the arrival corresponding to departure $m_2 \in \mathcal{M}$. The turnaround time $l_{m_{12}}$ is the time difference between the two movements

t_m : Requested time slot for movement $m \in \mathcal{M}$

c_m^t : Cost of allocating movement $m \in \mathcal{M}$ to time slot t, with $c_m^t = |t - t_m|$

$\alpha_a^{dt}, \beta_a^{dt}, \gamma_a^{dt}$: Departure, arrival and total capacity available at airport $a \in \mathcal{A}$ on day $d \in \mathcal{D}$ at time $t \in \mathcal{T}$.

w : Weight of the second-stage estimation, $w = 1 \Rightarrow$ each scheduled day has the same weight as the schedule, $w = \dfrac{1}{|\mathcal{D}|} \Rightarrow$ all days together have the same weight as the schedule

Decision Variables

$$x_m^t = \begin{cases} 1, & \text{if movement } m \in \mathcal{M} \text{ is allocated to time slot } t \in \mathcal{T} \\ 0, & \text{otherwise} \end{cases}$$

It is important to highlight that the number of variables is not related to the number of days on which movements have been requested. Therefore, if a single movement can be assigned to ten different time instants, the corresponding number of decision variables is equal to ten even if the movement is requested on a large set of days, as the time slot assigned is the same on each requested day.

The first-stage formulation can now be stated as follows:

$$\text{Min} \sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{T}} c_m^t x_m^t + w \cdot \mathbb{E}[f(x, \tilde{\omega})] \tag{2.1a}$$

$$\text{s.t.} \sum_{t \in \mathcal{T}} x_m^t = 1 \qquad \forall m \in \mathcal{M} \tag{2.1b}$$

$$\sum_{m \in D_a^d} \sum_{t \in [\tau, \tau + t_c)} x_m^t \leq \alpha_a^{d\tau} \qquad \forall a \in \mathcal{A}, c \in DC_a, d \in \mathcal{D}, \tau \in \mathcal{T} \tag{2.1c}$$

$$\sum_{m \in A_a^d} \sum_{t \in [\tau, \tau + t_c)} x_m^t \leq \beta_a^{d\tau} \qquad \forall a \in \mathcal{A}, c \in AC_a, d \in \mathcal{D}, \tau \in \mathcal{T} \tag{2.1d}$$

$$\sum_{m \in D_a^d \cup A_a^d} \sum_{t \in [\tau, \tau + t_c)} x_m^t \leq \gamma_a^{d\tau} \qquad \forall a \in \mathcal{A}, c \in TC_a, d \in \mathcal{D}, \tau \in \mathcal{T} \tag{2.1e}$$

$$x_{m_1}^t - x_{m_2}^{t + t_{m_{12}}} = 0 \qquad \forall (m_1, m_2) \in \mathcal{F},\ t \in \mathcal{T} \tag{2.1f}$$

$$\sum_{t \in [0, k)} x_{m_2}^t + \sum_{t \in [k - l_{m_{12}}, |\mathcal{T}|)} x_{m_1}^t \leq 1 \qquad \forall (m_1, m_2) \in \mathcal{P},\ k \in [l_{m_{12}}, |\mathcal{T}|) \tag{2.1g}$$

$$x_m^t \in \{0, 1\} \qquad \forall m \in \mathcal{M},\ t \in \mathcal{T} \tag{2.1h}$$

In the first-stage, a decision is made on the schedule of requested movements $m \in \mathcal{M}$ at every airport $a \in \mathcal{A}$, taking into account the effect that possible future capacity reductions – on the day of operation of flights – may have in terms of delays. The objective function (2.1a) minimizes the sum of the schedule/request discrepancies and operational delays, with the w parameter taking care of properly balancing the first- and second-stage decisions. This is very important when a large number of days are considered for scheduling: in such case, $w = 1$ gives little importance to airline requests, favoring the reduction of delays on the day of operation. The recourse function $f(x, \omega)$ is given as defined in the following of this section. Constraints (2.1b) guarantee that a time slot is allocated to every requested movement. Constraints (2.1c), (2.1d) and (2.1e) ensure that departure, arrival and total capacity constraints are satisfied at every airport $a \in \mathcal{A}$ all the time. Constraints (2.1f) guarantee that the flight time for all flights is fixed, by properly distancing departure movements from the corresponding arrivals. The two decision variables referring to the departure movement at time t and to the arrival movement at time $t + t_{m_{12}}$ of a flight have the same value, as they are either both be equal to 0 (flight not departing at t) or to 1 (flight departing at t, therefore arriving at $t + t_{m_{12}}$). Constraints (2.1g) guarantee that the turnaround time for all aircraft is respected, by properly distancing all couples of consecutive movements $(m_1, m_2) \in P$. Finally, constraints (2.1h) impose binary restrictions on the x_m^t decision variables.

Formulation of the second-stage

The second-stage value function that measures the expected delays – assigned to flights in order to meet airport capacity realizations – taken over all the possible realizations of the random event ω is now described. In what follows, uncertainty is supposed to only affect capacity at airports. More specifically, the right-hand side of the capacity constraints depends on the random variable $\tilde{\omega}$. This value function does not provide the exact amount of delay per each scenario, but rather it determines the number of delayed departure and arrival movements resulting from a specific capacity realization. This is done macroscopically, without considering the effects of delays on specific flights on the complete network, in order to have second-stage formulations that are easy to compute.

Decision Variables

y^{ω}_{adt} : Number of delayed departures from airport $a \in \mathcal{A}$ on day $d \in \mathcal{D}$ at time instant $t \in \mathcal{T}$ for capacity realization $\omega \in \Omega$

z^{ω}_{adt} : Number of delayed arrivals to airport $a \in \mathcal{A}$ on day $d \in \mathcal{D}$ at time instant $t \in \mathcal{T}$ for capacity realization $\omega \in \Omega$

Given a first-stage decision – which is the schedule of departures and arrivals at each airport and for each time instant – and the realization of the random event ω – which is the capacity at each airport and for each period of time – a certain number of flights has to be delayed to avoid imbalances between demand and capacity. For each airport $a \in \mathcal{A}$, time instant $t \in \mathcal{T}$ and day $d \in \mathcal{D}$, defining $T^t_c = \{k | t \leq k < t + t_c\}$, the set of feasible second-stage decisions $\mathbb{Y}^t_{ad}(x, \omega)$ can be defined as follows:

$$
\begin{aligned}
\mathbb{Y}^t_{ad}(x,\omega) = \{(y^{\omega}_{adt}, z^{\omega}_{adt}) \quad | \quad & y^{\omega}_{adt} \in \mathbb{R}_+, \quad z^{\omega}_{adt} \in \mathbb{R}_+, \\
& y^{\omega}_{adt} \geq \sum_{m \in D^d_a, \tau \in T^t_c} x^{\tau}_m - \alpha^{dt}_a(\omega) && \forall c \in DC_a, \\
& z^{\omega}_{adt} \geq \sum_{m \in A^d_a, \tau \in T^t_c} x^{\tau}_m - \beta^{dt}_a(\omega) && \forall c \in AC_a, \\
& y^{\omega}_{adt} + z^{\omega}_{adt} \geq \sum_{m \in D^d_a \cup A^d_a, \tau \in T^t_c} x^{\tau}_m - \gamma^{dt}_a(\omega) && \forall c \in TC_a\}
\end{aligned}
$$

The stochastic program has complete recourse, therefore $\mathbb{Y}^t_{ad}(x, \omega) \neq \emptyset$ for any x and realization ω. Denoting with q_a the cost for delaying a movement at airport $a \in \mathcal{A}$, the second-stage value function is given by:

$$
f(x,\omega) = \text{Min} \sum_{a \in \mathcal{A}} q_a \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} (y^{\omega}_{adt} + z^{\omega}_{adt}) \tag{2.2a}
$$

$$
(y^{\omega}_{adt}, z^{\omega}_{adt}) \in \mathbb{Y}^t_{ad}(x,\omega) \qquad \forall a \in \mathcal{A}, \; d \in \mathcal{D}, t \in \mathcal{T}. \tag{2.2b}
$$

The second-stage value function described above is only an estimate of the delays assigned to movements in order to meet available capacity given slot assignment x. Indeed, this second-stage formulation penalizes the excess of demand at each time period without taking into account the interactions between time periods, i.e., it does not capture the propagation of delays from one time period to the next. Delay propagation results in having a delayed arrival (departure) movement at time instant t increase the demand of arrivals (departures) at time instant $t+1$ by one unit. Due to this simplification in the formulation, this formulation is called simplified recourse. To overcome this issue, a different formulation of the second-stage is now considered, in which demand at consecutive time instants is linked, i.e., movements not operated at time t increase the demand for movements at time $t+1$. This provides a better estimate of delays. Due to its nature, this formulation is referred to as time-linked recourse.

$$f(x,\omega) = \text{Min} \sum_{a\in\mathcal{A}} q_a \sum_{d\in\mathcal{D}} \sum_{t\in\mathcal{T}} (y^{\omega}_{adt} + z^{\omega}_{adt}) \tag{2.3a}$$

$$(y^{\omega}_{adt} - y^{\omega}_{ad,t-1}, z^{\omega}_{adt} - z^{\omega}_{ad,t-1}) \in \mathbb{Y}^{t}_{ad}(x,\omega) \qquad \forall a \in \mathcal{A},\ d \in \mathcal{D}, t \in \mathcal{T}. \tag{2.3b}$$

The better quality of the time-linked recourse function comes at the cost of a higher computational burden. In fact, the second-stage program cannot be decomposed into subproblems for each time instant t, as in the case of the simplified recourse. Subproblems may however be defined for each airport a and schedule day d. Furthermore, it is important to notice that neither recourse formulation takes the effect of delay on a single flight into account, which prevents from evaluating the network effect of delays at this stage. This is due to the fact that both formulations are macroscopic in nature to allow for relatively fast computation times.

2.2.3 Second-stage formulation example

A practical example to provide a better understanding of the two second-stage formulations is now provided, highlighting the differences between them. Consider a schedule x (first-stage solution) and a capacity realization ω. The proposed schedule corresponds to a specific demand d^{dt}_a for departures and a^{dt}_a for arrivals at each considered airport $a \in \mathcal{A}$, scheduled day $d \in \mathcal{D}$, and time instant $t \in \mathcal{T}$. Capacity defined at each time instant is also considered, affecting the same time instant only, i.e., the $\alpha^{dt}_a(\omega), \beta^{dt}_a(\omega)$, and $\gamma^{dt}_a(\omega)$ values define the maximum demand that can be accommodated at each time instant t. In the example illustrated in Table 2.2, it is possible to observe how the objective values of the simplified and time-linked recourse formulations can be significantly different.

In Table 2.2, the first row (Time) provides the different time instants considered. The second row (d^{dt}_a) shows the instant departure demand at the considered airport a and day d. Similarly, the third row (α^{dt}_a) depicts available departure capacity at

Table 2.2: Second-stage delay calculation example

Time	1	2	3	4	5	6
d_a^{dt}	5	6	7	6	6	5
α_a^{dt}	6	6	6	6	6	6
Simpl. delay	0	0	1	0	0	0
T-link. delay	0	0	1	1	1	0

each time instant. For the sake of simplicity, arrival and total capacity constraints are not considered. Finally, rows four and five provide the amount of delay estimated by each recourse formulation. In this table, it is possible to observe that at time $t = 3$ capacity is exceeded by one departure movement. Demand matches capacity at the two following time instants, therefore this departure delay has a downstream effect, causing delays at time instants 4 and 5 too. This fact is correctly captured by the time-linked recourse formulation, which has an objective value for this part of the problem equal to 3. The simplified recourse formulation, on the other hand, only detects on how many time instants capacity is exceeded, without considering the downstream effect of delays, thus providing an objective value equal to 1.

2.3 Sample Average Approximation

Two-stage stochastic programming models with a general formulation discussed in §1.4, such as the TSAPU formulations presented in §2.2, can be decomposed with stage-wise decomposition methods. This involves solving the first-stage of the problem first, and then evaluating the proposed solution over the set of scenarios Ω. However, the number of possible scenarios may be very large, especially when different independent phenomena are considered. This is the case for the instances of the TSAPU, as weather at different airports is considered independently, since the problem can be solved for a network of airports that are very distant one from another. Also, the correlation between weather conditions on different days of the calendar horizon to schedule is not considered, i.e., weather conditions are considered independently on each day. This contributes to the further increase of the number of scenarios to be considered. Therefore, the number of possible scenarios for each instance of the TSAPU is given by:

$$|\Omega| = \prod_{a \in \mathcal{A}} \Omega_a^{|\mathcal{D}|} \tag{2.4}$$

where Ω_a is the number of possible capacity realizations at airport $a \in \mathcal{A}$ and $|\mathcal{D}|$ is the number of scheduled days. This number increases very quickly with the number of airports $|\mathcal{A}|$, scheduled days $|\mathcal{D}|$, and capacity realizations at each airport Ω_a.

For such problems, evaluating each possible scenario is not computationally feasible. Therefore, sampling methods need to be used. To solve instances of the TSAPU, the SAA method [39] was implemented in C++ Language, using the CPLEX 12.1 Callable Libraries [40]. SAA is based on the fact that the objective function of the problem involves the expected value of a random variable. In fact, since $\tilde{\omega}$ is a random variable, $f(x, \tilde{\omega})$ is a random variable for each $x \in X$ too. Therefore, $E[f(x, \tilde{\omega})]$ can be estimated statistically. SAA uses exterior sampling, i.e., sampling is performed outside of the algorithm and does not depend on its execution.

Given $N \in \mathbb{N}$, let $\{\omega^t\}_{t=1}^N$ be a random sample of size N from $\tilde{\omega}$, where the ω^t values are independent and identically distributed (i.i.d.) observations of $\tilde{\omega}$. Replacing $\tilde{\omega}$ with $\tilde{\omega}_N$ – which has a distribution that is the empirical distribution of $\{\omega^t\}_{t=1}^N$ – it is possible to estimate $E[f(x, \tilde{\omega})]$ with the following stochastic linear problem, called SAA problem:

$$\begin{aligned} \hat{z}_N = \operatorname{Min} c^\top x + \frac{1}{N} \sum_{t=1}^{N} f(x, \omega^t) \\ \text{s.t. } Ax \geq b \\ x \in X. \end{aligned} \tag{2.5}$$

The optimal solution x_N^* to problem (2.5) can be determined using any optimization algorithm. The implemented code uses the L-Shaped algorithm to deal with the N subproblems that derive from the chosen sample. This solution is not necessarily a valid solution to the original problem (1.1). In fact, it is a valid solution for the considered sample. This fact can be extended to the original problem if the problem has (relatively) complete recourse. This is the case of the TSAPU formulations proposed, therefore x_N^* is a feasible solution to the original problem. However, it is not necessarily the optimal solution, as optimality is guaranteed for the chosen sample only.

Problem (1.1) has an optimal solution x^* that corresponds to the optimal objective value $\hat{z}$. With SAA, it is possible to provide both statistical lower and upper bound limits to the z^* value. The lower bound can be obtained observing that the optimal solution x_N^* to a problem solved with a sample of size N provides, by definition, an objective value that is at least as good as that provided by any other solution, including the optimal solution x^* to the problem that considers the whole set of scenarios Ω, i.e.,

$$c^\top x_N^* + E[f(x_N^*, \tilde{\omega}_N)] \leq c^\top x^* + E[f(x^*, \tilde{\omega}_N)] \tag{2.6}$$

Taking the expectation on both sides, the previous inequation becomes:

$$E[\hat{z}_N] \leq \hat{z} \tag{2.7}$$

Inequation (2.7) indicates that the optimal objective value $\hat{z}$ of problem (1.1) is lower bounded by the expectation of the optimal objective value of problems solved for a sample of size N. To get this expectation, it is necessary to solve M different problems with sample of size N, i.e., the SAA problem must be solved on M different samples. This corresponds to the following:

$$E[\hat{z}_N] = \frac{1}{M}\sum_{i=1}^{M} \hat{z}_N^i \tag{2.8}$$

where $\hat{z}_N^i$ is the optimal objective value of the problem solved on the i-th sample, i.e.,

$$\begin{aligned} \hat{z}_N^i = \text{Min}\, c^\top x &\sum_{t=1}^{N} \frac{1}{N} f(x, \omega^{t,i}) \\ \text{s.t. } Ax &\leq b \\ x &\in X \end{aligned} \tag{2.9}$$

In practice, since the implemented code uses the L-Shaped algorithm to solve a problem with a sample of size N, estimating $\hat{z}_N$ requires M executions of the L-Shaped algorithm. Once these computations are performed, it is possible to compute the $(1-\alpha)$ confidence interval for the lower bound. Denoting with LB the average objective value obtained by the M replications, the Central Limit Theorem indicates the following convergence in distribution:

$$\sqrt{M}(LB - E[\hat{z}_N]) \Rightarrow N(0, Var(\hat{z}_N)) \quad \text{for } M \to \infty \tag{2.10}$$

It is also necessary to define the lower bound sample variance estimator, which is given by:

$$S_L^2(M) = \frac{1}{M-1}\sum_{i=1}^{M} (\hat{z}_N^i - LB)^2 \tag{2.11}$$

Defining z_α s.t. $P\{N(0,1) \leq z_\alpha\} = 1 - \alpha$, it is possible to define the $(1-\alpha)$ confidence interval for the lower bound as follows:

$$\left[LB - \frac{z_{\alpha/2} \cdot S_L(M)}{\sqrt{M}}, LB + \frac{z_{\alpha/2} \cdot S_L(M)}{\sqrt{M}}\right] \tag{2.12}$$

After defining the lower bound, it is also possible to define an upper bound to the optimal solution value of the problem. The upper bound can be determined observing that any feasible solution to the problem, including the N SAA problem solutions x_N^{i*}, with $i \in [1, N]$, are at most as good as the optimal solution to the original problem calculated on the whole set of scenarios Ω. That is,

$$\hat{z} \leq c^\top x_N^{i*} + E[f(x_N^{i*}, \tilde{\omega})] \tag{2.13}$$

It is not possible to exactly compute the right-hand side because it would require to compute function f for all scenarios $\omega \in \Omega$. Therefore, it needs to be statistically estimated. To do so, it is necessary to generate $\overline{M}$ i.i.d. samples of size $\overline{N} >> N$, i.e., $\{\omega^{t,j}\}_{t=1}^{\overline{N}}$ for $j \in [1, \overline{M}]$. The objective value associated with each sample j is defined by:

$$\hat{z}_{\overline{N}}^j(x_N^{i*}) = c^\top x_N^{i*} + \frac{1}{\overline{N}} \sum_{t=1}^{\overline{N}} f(x_N^{i*}, \omega^{t,j}) \tag{2.14}$$

The estimation of the upper bound, calculated for the feasible solution x_N^{i*}, is defined as follows:

$$UB(x_N^{i*}) = \frac{1}{\overline{M}} \sum_{j=1}^{\overline{M}} \hat{z}_{\overline{N}}^j(x_N^{i*}) \tag{2.15}$$

To formulate the confidence interval of the upper bound, it is necessary to define the upper bound sample variance estimator as follows:

$$S_U^2(\overline{M}, x_N^{i*}) = \frac{1}{\overline{M}-1} \sum_{j=1}^{\overline{M}} \left(\hat{z}_{\overline{N}}^j - UB(x_N^{i*})\right)^2 \tag{2.16}$$

Finally, with z_α defined as for the lower bound, it is possible to formulate the $(1-\alpha)$ upper bound confidence interval, calculated using solution x_N^{i*}:

$$\left[UB(x_N^{i*}) - \frac{z_{\alpha/2} \cdot S_U(\overline{M}, x_N^{i*})}{\sqrt{\overline{M}}}, UB(x_N^{i*}) + \frac{z_{\alpha/2} \cdot S_U(\overline{M}, x_N^{i*})}{\sqrt{\overline{M}}}\right] \tag{2.17}$$

The upper bound allows to determine the quality of a solution x_N^{i*} chosen from the N SAA replications, compared with the lower bound that indicates how good the optimal solution could be. Confidence intervals also allow to highlight the uncertainty on these bounds: the narrower the intervals, the more precise they are. In the following section, SAA lower and upper bounds are used to analyze the solutions to different instances of the TSAPU.

2.4 TSAPU Computational Results

In this section, the computational experiments performed using the mathematical model described in §2.2 are presented. Ten realistic problem instances were generated following real air traffic profiles from different airports. Each instance represents a different network of interconnected airports, and considers four different scheduled days. Due to the small number of days considered, for all computational experiments the first-stage objective function parameter w is set to 1, i.e., the schedule/request discrepancies have the same weight as the sum of expected delays on all four scheduled days. Each day is subdivided into 5' wide time instants. Nominal airport capacity is based on real data – where it was possible to retrieve this information – or generated from the analysis of airport schedules. Considered capacity constraints apply to departure, arrival and total flight movements, and they all have a one-hour duration. For example, a total capacity constraint of 20 movements at airport a and at the time instant corresponding to 10:00 imposes a limitation of 20 total movements (arrivals and departures) between 10:00 and 11:00. Capacity reduction scenarios are generated from real historical weather conditions at the different airports. Data on weather conditions spanning over one month were analyzed to define the different scenarios. A single scenario represents a specific capacity realization at each different airport for the whole scheduling horizon. Flight movement requests are generated from available real airport schedules. This generation involves perturbing the time of departure and of arrival of certain flights in order to simulate the real requests of airlines prior the assignment of time slots. The perturbation consists in shifting both the time of departure and of arrival of selected flights either one time period in the past or in the future. Selected flights are those operated at "appealing time slots", i.e., time slots of periods of time at which an airport operates at its nominal capacity. Flight connectivity is not implemented due to the lack of real information.

In the following, the ten considered instances are described. A list with the name and International Civil Aviation Organization (ICAO) code for all airports included in each instance is provided. The list of airports for each instance is preceded by an abbreviated name with which instances are referred to in the following of this chapter.

- *2europe*: Munich (EDDM), Toulouse (LFBO)
- *3europe*: Munich (EDDM), Venice Marco Polo (LIPZ), Paris Charles de Gaulle (LFPG)
- *4europe*: Lyon (LFLL), Helsinki-Vantaa (EFHK), Venice Marco Polo (LIPZ), Warsaw (EPWA)
- *5europe*: Hamburg (EDDH), London Gatwick (EGKK), Copenhagen (EKCH), Lisbon (LPPT), Basel-Mulhouse-Freiburg (LFSB)
- *6europe*: Hamburg (EDDH), Edinburgh (EGPH), Liverpool (EGGP), Warsaw (EPWA), Barcelona (LEBL), Porto (LPPR)
- *west*: Paris Charles de Gaulle (LFPG), Barcelona (LEBL), Milan Linate (LIML)
- *southwest*: Rome Fiumicino (LIRF), Barcelona (LEBL), Lyon (LFLL), Porto (LPPR)
- *france*: Paris Charles de Gaulle (LFPG), Toulouse (LFBO), Lyon (LFLL), Basel-Mulhouse-Freiburg (LFSB)
- *italy*: Rome Fiumicino (LIRF), Milan Linate (LIML), Catania (LICC), Palermo (LICJ), Cagliari (LIEE)
- *large*: Bergamo (LIME), Beauvais Tillé (LFOB), London Gatwick (EGKK), Tarbes-Lourdes (LFBT), Rhodes International (LGRP), Menorca (LEMH), Porto (LPPR), East Midlands (EGNX), Pisa (LIRP), Ibiza (LEIB)

Additional details on the size of each instance are provided in Table 2.3. Column 1 (Instance) provides the name of each instance. Columns 2 (Flights) and 3 (Mov.s) provide the number of flights and movements considered for each instance, respectively. Notice that each flight corresponds to a couple of movements, and movements can be operated on multiple days, resulting in a number of movements smaller than that of flights in the table. Columns 4 (1st Cols) and 5 (1st Rows) provide the size of first-stage problems, while columns 6 (2nd Cols) and 7 (2nd Rows) provide the size of (second-stage) subproblems. Subproblems have the same size with the simplified and time-linked recourse formulations. Finally, column 8 ($|\Omega|$) provides the total number of possible scenarios for each instance, calculated with formula (2.4).

Furthermore, some insight on the level of congestion presented by each instance is provided in Table 2.4. In column 2, the average number of movement requests per airport

Table 2.3: Description of the instances

Instance	**Flights**	**Mov.s**	**1st Cols**	**1st Rows**	**2nd Cols**	**2nd Rows**	$\lvert\Omega\rvert$
2europe	5,675	3,392	42,648	6,116	2,280	4,170	$3.5 \cdot 10^{7}$
3europe	10,795	6,492	82,332	13,061	3,468	8,428	$3.7 \cdot 10^{6}$
4europe	5,969	3,724	46,919	12,952	4,632	10,570	$8.5 \cdot 10^{7}$
5europe	9,267	6,402	81,057	15,716	5,760	10,939	$5.5 \cdot 10^{11}$
6europe	8,458	5,872	74,057	13,917	6,912	10,119	$6.9 \cdot 10^{13}$
west	9,962	6,010	76,100	10,539	3,468	6,312	$1.7 \cdot 10^{6}$
southwest	9,058	5,712	72,447	11,591	4,620	7,539	$5.0 \cdot 10^{11}$
france	8,474	5,288	68,010	11,209	4,673	6,366	$6.5 \cdot 10^{10}$
italy	5,586	3,530	49,676	21,920	5,812	14,303	$2.5 \cdot 10^{9}$
large	6,520	5,702	73,068	25,594	11,554	20,423	$1.4 \cdot 10^{17}$

that exceed available "instant capacity" is provided. The term "instant capacity" refers to the result of the subdivision of total capacity, which is defined for a set of contiguous time instants (in the considered case, the set spans over a one-hour horizon), over single time instants, that are five-minute wide. Column 3 provides the average number of time instants per airport at which demand is greater or equal to the instant capacity.

2.4.1 Experimental results

All instances were solved using both recourse formulations on a computer with Intel Xeon E5620 dual core CPU at 2.40GhZ with 12 GB of RAM on a 64-bit Windows 7 operating system. The model was solved using the SAA method described in §2.3, implemented in C++ language using the CPLEX 12.1 libraries. To compute the lower bound, 20 SAA replications were performed for each instance ($M = 20$), using different samples of size 100 ($N = 100$). The SAA optimality gap varies between 1% and 5% depending on

Table 2.4: Instance congestion

Instance	**Exceeding demand/airport**	**Congested instants/airport**
2europe	129.0	270.5
3europe	123.3	250.3
4europe	109.3	86.8
5europe	203.6	196.0
6europe	140.2	99.2
west	72.0	98.0
southwest	86.5	100.8
france	74.3	77.8
italy	469.8	174.6
large	375.0	189.2

the instance. For instances taking a longer computation time, the gap was set to 5%, while for "easy" instances the gap was set to 1%. To compute the upper bound, 20 SAA replications were performed ($\overline{M} = 20$), using samples of size 1000 ($\overline{N} = 1000$).

The computational results for the simplified and time-linked recourse formulations are reported in Tables 2.5 and 2.6, respectively. The two tables share the following structure. Column 1 (Instance) provides the name of each instance. Column 2 (LB) shows the 95% confidence interval for the lower bound. The solution out of the 20 SAA replications whose value is the closest to the average value computed by the SAA is then chosen to compute the upper bound value, whose 95% confidence interval is reported in column 3 (UB). Similarly, column 4 reports the 95% confidence interval of the upper bound computed using the solution of the problem that considers nominal capacity, i.e., without considering uncertainty. When values in column 3 are lower than values in column 4 (Nom. UB), the TSAPU model provides an improvement over a model that does not consider uncertainty. Columns 5 (1st N.) and 6 (1st U.) compare the first-stage objective values, i.e., the schedule/request discrepancies, of the solution computed only considering nominal capacity (column 5) and of the SAA solution chosen to compute the upper bound (column 6). The larger the difference between these two values, the more different the schedules proposed with and without considering uncertainty. Finally, the last two columns (LB t (h) and UB t (h)) provide the computation times for the lower and upper bounds. Since computing the upper bound involves computing a large number of subproblems that consider different scenarios, the upper bound computation times reflect the level of difficulty of solving the subproblems. Comparing the values in the two tables, it is possible to observe that it is equally difficult to solve subproblems with simplified or time-linked recourse. However, simplified recourse subproblems may be decomposed by day, airport, and time into smaller independent problems, while time-linked recourse subproblems cannot be decomposed by time. Therefore, implementing parallel algorithms it may be possible to achieve lower solution times with the simplified formulation compared with the time-linked formulation. Lower bound computation times, on the other hand, can differ greatly between simplified and time-linked recourse due to the different cuts provided by subproblems. Averaging over the considered instance set, calculating the lower bound appears to be faster using the simplified recourse formulation, which takes 4.6h on average, against 5.2h needed by the time-linked recourse formulation. However, this is an instance-dependent factor. For example, the lower bound for instance *italy* is computed in 8.1h using the simplified recourse formulation, while it only takes 5h to be computed using the time-linked formulation.

Table 2.5: Simplified recourse results

Instance	LB	UB	Nom. UB	1st N.	1st U.	LB t (h)	UB t (h)
2europe	[364.0, 388.7]	[385.3, 391.5]	[378.1, 386.5]	177	194	1.0	0.7
3europe	[403.5, 428.4]	[422.9, 429.5]	[430.9, 439.4]	177	202	6.2	1.8
4europe	[43.3, 44.2]	[45.6, 45.8]	[62.2, 62.8]	21	35	3.8	2.5
5europe	[361.1, 365.3]	[369.6, 371.0]	[399.0, 400.2]	282	305	5.2	2.9
6europe	[85.6, 95.6]	[91.4, 94.4]	[93.7, 96.5]	0	1	1.0	2.4
west	[115.0, 137.9]	[127.9, 136.2]	[129.5, 137.5]	0	0	0.2	1.1
southwest	[105.3, 109.7]	[106.3, 107.8]	[123.9, 125.6]	35	46	4.1	1.4
france	[45.0, 45.7]	[45.3, 45.5]	[92.7, 93.3]	3	32	13.3	1.3
italy	[468.8, 518.1]	[466.1, 484.0]	[526.8, 545.2]	41	108	8.1	5.4
large	[726.5, 739.7]	[728.0, 731.2]	[736.7, 741.4]	608	612	3.7	10.1

Table 2.6: Time-linked recourse results

Instance	LB	UB	Nom. UB	1st N.	1st U.	LB t (h)	UB t (h)
2europe	[620.9, 684.5]	[652.7, 668.2]	[685.5, 711.6]	177	274	4.4	0.6
3europe	[675.0, 735.0]	[739.2, 753.7]	[745.7, 770.3]	177	344	10.7	1.9
4europe	[43.3, 44.3]	[43.7, 43.9]	[62.2, 62.8]	21	32	3.9	2.5
5europe	[371.9, 377.4]	[374.5, 376.5]	[410.6, 412.4]	282	316	4.9	2.9
6europe	[184.2, 214.8]	[201.9, 215.8]	[206.9, 219.1]	0	1	0.7	2.5
west	[524.4, 622.3]	[581.8, 618.3]	[612.8, 654.0]	0	70	1.2	1.1
southwest	[114.6, 119.3]	[117.9, 119.9]	[133.3, 135.4]	35	44	4.0	1.4
france	[45.0, 45.6]	[50.5, 50.7]	[92.7, 93.3]	3	40	12.7	1.3
italy	[2,978.5, 3,370.1]	[2,896.7, 3,037.1]	[3,138.3, 3,354.0]	41	487	5.0	5.4
large	[984.0, 1,046.4]	[1,024.0, 1,038.5]	[1,030.1, 1,052.1]	608	644	4.1	10.0

2.4.2 Results interpretation

The utilization of the described TSAPU model leads to the proposal of a tradeoff between schedule quality, i.e., the schedule/request discrepancies, and future delays. The difference between columns 6 (1st U.) and 5 (1st N.) from Tables 2.5 and 2.6 corresponds to the “cost” of the proposed new schedule. In fact, column 6 shows the schedule/request discrepancies of the proposed schedule, while column 5 shows the schedule/request discrepancies of the optimal schedule, which is the minimum possible value. This tradeoff is analyzed in Table 2.7. Here, in columns 2 and 5 the “price” to pay in terms of first-stage delay for the simplified and time-linked recourse formulations is shown. Paying this “price” it is possible to reduce future demand-capacity imbalances. This benefit is evaluated by comparing the second-stage cost estimated by using the nominal solution, which guarantees schedule optimality, with that estimated by using the solution computed with SAA. The value of this benefit is shown in columns 3 and 6. The decision maker can evaluate this tradeoff and decide whether to keep the schedule that only minimizes the schedule/request discrepancies or to opt for a more robust one, that allows to save delay on the day of operation of flights. The percentage of global gain in terms of objective value that arises from using the schedule proposed by solving the proposed model is also shown in columns 4 and 7.

The global gain that can be achieved with the discussed tradeoff differs greatly instance by instance. With the simplified recourse formulation it is possible to provide improvements in nine out of ten cases. Using the time-linked recourse formulation, which is more precise than the simplified recourse formulation, it is possible to provide improvements for all considered instances. However, the degree of improvement can range from very limited (see *large*, *3europe* and *6europe*, where improvements do not exceed 2%) to very relevant (see *southwest*, *4europe* and *france*, where improvements are over 10%, with the last instance showing an improvement of over 45%).

The simplified and time-linked formulations lead to schedules that can be very different. For example, consider instance *west*. Using the simplified recourse formulation one would choose an optimal schedule, different from the originally selected one (there are multiple optimal schedules), that provides some future benefit. On the other hand, using the time-linked recourse formulation, the first-stage objective increases from 0 to 70, indicating a much different proposed schedule. This results in a larger future benefit, as can be observed in Table 2.7: there are 103.3 fewer capacity violations on the day of operation, compared with 1.4 fewer capacity violations obtained using simplified recourse.

This analysis can be seen as a method that can be applied to suggest possible improvements that can make a schedule more robust. In particular, Table 2.7 constitutes a good instrument for evaluating the proposed results, indicating different possible actions from case to case.

Table 2.7: Delay tradeoff

	Simplified recourse			**Time-linked recourse**		
Instance	**1st Add.**	**2nd Saved**	**Global gain**	**1st Add.**	**2nd Saved**	**Global gain**
2europe	17	10.9	-1.6%	97	135.1	5.5%
3europe	25	33.9	2.1%	167	178.5	1.5%
4europe	14	30.8	26.9%	11	29.7	29.9%
5europe	23	52.3	7.3%	34	70.0	8.7%
6europe	1	2.9	2.0%	1	5.2	2.0%
west	0	1.4	1.1%	70	103.3	5.3%
southwest	11	28.7	14.2%	9	24.5	11.5%
france	29	76.6	51.2%	37	79.4	45.6%
italy	67	128.0	11.4%	446	725.3	8.6%
large	4	13.5	1.3%	36	45.8	0.9%

2.5 TSAPU Conclusions

In this chapter, a two-stage stochastic programming model for the TSAPU with two alternative formulations was presented. This model has the goal of determining robust flight schedules, i.e., schedules that consider both airline requests for time slots and future delay on the day of operation that can be expected by using a specific schedule.

Today, slot allocation practice suffers several inefficiencies. Slot coordinators have to make complex decisions with limited support from rule-driven slot management applications. These decisions are made locally, despite having a relevant network effect, as assigning a departure time slot at some airport needs to correspond to the assignment of a specific arrival time slot at another airport. Local decisions are currently coordinated with biannual conferences, which constitute a big overhead in the decision making process. Furthermore, the regulatory framework on slot allocation needs to be improved to achieve a more efficient use of scarce capacity, notably through the use-it-or-lose-it rule, the new entrant rule, revenue neutral measures and secondary trading, see [41].

The proposed TSAPU model constitutes a new decision support tool for slot coordinators, as they can use it to assign time slots according to airline requests and the future impact of the schedule. Two alternative formulations which consider multiple airports at the same time are proposed, thus taking into account the complex network interdependency that is not considered when schedules are proposed locally.

The contribution to literature of this work is in providing a model that, to the best of the author's knowledge, is the first model for time slot allocation that considers multiple airports being scheduled as a whole network and the uncertainty on future capacity realizations. The model does not implement any of the existing IATA rules for time slot allocation (e.g., it does not consider different priority rules for flights) and is based on optimization criteria only. It is however possible to easily align it with the existing IATA slot allocation procedures by applying the model sequentially to each priority class, i.e., first allocating requests holding "grandfather rights", then allocating 50% of the new capacity, including withdrawn and surrendered slots, to requests of "new entrants", and finally allocating all remaining time slot requests. This would lead to much faster solutions for each subproblem than the times reported in §2.4, as each of them would be of smaller dimensions.

The practical impact of the discussed model is that it can be a useful tool for slot coordinators to evaluate the tradeoff between the distance of the schedule from airline requests and future delays. In fact, by paying a "cost" given by an increased distance of the schedule from airline requests, it is possible to obtain a reduction of delays that are expected to occur on the day of operation of the flights. As highlighted in §2.4, this tradeoff can range from very limited to highly beneficial. In one of the instances, it was possible to reduce total delay costs, i.e., the sum of the schedule/request discrepancies and operational delays, by over 45% using the time-linked recourse formulation, which is the most precise formulation proposed. It should also be noted that another possible interpretation of the results is determining the optimal capacity levels at each considered airport that minimize the sum of schedule/request discrepancies and operational delays.

The computation times obtained are viable for the application, as they are consistent with the time constraints associated with the current decision making interaction between slot coordinators and airlines, which is a process spanning over several days. Good results and computational viability together show that the proposed approach is very promising and deserves further investigation, as it may lead to relevant monetary benefits for airlines and all other stakeholders. Future research will first focus on the analysis of the fair distribution of the schedule/request discrepancies among flights. This involves changing the objective function to include fair time slot allocation among different flights, to avoid over-penalizing a small set of flights while favoring other flights. This analysis should then be extended to fair allocation among airlines, however this development depends on the availability of complete real flight data that were not available for this study. Also, these promising results should be studied further, involving wider networks of coordinated airports and considering a larger scheduling horizon. Both factors are critical as they will make the problem of larger scale, however the discussed subdivision of the problem into smaller subproblems considering "grandfather rights" from IATA rules will help mitigate this increase in the size of the problem to solve. Future research will therefore be focused on showing that such problems are also solvable within acceptable computation times.

Chapter 3

TACTICAL DECISION MAKING

Most air traffic works available in literature face decision problems at the tactical phase, which takes place on the day of operation of flights. Different decisions can be made considering heterogeneous factors. Two main problems that involve flight schedules at this phase can be identified: the Ground Holding Problem (GHP) and the Air Traffic Flow Management (ATFM) problem. The GHP is typically faced either as the Single-Airport GHP (SAGHP) or the Multi-Airport GHP (MAGHP). The GHP only defines ground holding policies at different airports to regulate air traffic. ATFM, on the other hand, usually defines the whole flight plan, with ground holding being one of the considered control options. The definition of complete flight plans is important when the airspace may be congested, which applies to the European case more than to the U.S. case. However, since solutions are more into detail, ATFM models are more difficult to solve than GHP models. For example, computational results for a ground holding model proposed in 1998 by Andreatta *et al.* [42, 43] report computation times of less than 20' for instances with over 20,000 flights, while it is still very challenging to solve instances that consider the same number of flights with ATFM models within similar computation times nowadays. A review of models for these tactical air traffic problems is provided by Agustín *et al.* [44], in which both SAGHP and MAGHP models are described, as well as deterministic and stochastic models for the ATFM problem.

Two different problems that arise at the tactical phase are faced in this chapter. The first problem involves ATFM considering the uncertainty on capacity availability. This is the same kind of uncertainty considered in the strategic phase, but uncertainty is now much lower, i.e., the number of possible scenarios is limited. This problem is called Stochastic Air Traffic Flow Management (SATFM) problem, and is discussed in §3.1. A stochastic optimization model to solve it is proposed in §3.2, and an ad-hoc heuristic to solve its instances is described in §3.3. In §3.4, the results are shown for the computational experiments performed on a set of realistic instances, and conclusions are discussed in §3.5.

The second tactical problem faced also involves ATFM. Differently from the problems faced so far, it does not consider the uncertainty on capacity availability. The uncertainty considered by the second problem is the implicit uncertainty on the time of departure of flights, i.e., the fact that a flight may not be able to depart at its scheduled time due to operational problems. To solve this problem, time windows are used, and the problem is therefore called the Air Traffic Flow Management problem with Time Windows (ATFMTW). This problem is illustrated in §3.6, and a deterministic optimization model to solve it is presented in §3.7. The model is solved using a standard optimization solver, and it was tested on a set of simulated instances. The results of the computational experiments are illustrated in §3.8. Finally, the conclusions and future directions of research are discussed in §3.9.

3.1 The Stochastic Air Traffic Flow Management Problem

The need to operate a growing number of flights is not currently accompanied by a sufficient increase of the resources needed to operate them. This fact is at the origin of congestion at airports and in the airspace, especially terminal airspace around airports. The increasing gap between demand for operations and the availability of resources leads to the need for the air traffic industry to manage its operations efficiently. ATFM faces the problem of regulating air traffic in order to avoid congestion. It is a typical problem that arises at the tactical phase, when flight plans need to be defined. ATFM decisions are typically not revised once a flight departs from the airport and enters the air traffic system.

The ATFM problem is widely faced in literature, considering a lot of different aspects. In particular, it is possible to subdivide ATFM models into two categories: deterministic and stochastic models. In this chapter, both kinds of models are formulated. First, in the current section, uncertainty on capacity availability is explicitly considered, identifying a set of possible different capacity realizations. To do so, a stochastic programming model that captures and represents the network effect of the ATFM problem is developed and implemented. Then, in §3.6, a different ATFM problem is faced, which is formulated with deterministic programming. Due to the difficulty of solving complex stochastic problems within short computational times, as required by ATFM problems, most related models do not consider any stochastic aspect. Stochastic models for the ATFM problem typically present some simplification in their formulation in order to keep the solution times low.

A review of the literature on stochastic models for the ATFM and ground holding problems is now presented. An early stochastic model of the ground holding problem in air traffic control was proposed by Richetta and Odoni [45]. This model presents the multi-period dynamic stochastic ground holding problem for a single destination airport, providing a dynamic multi-stage stochastic integer programming formulation with recourse actions. Ground delay decisions are made at each stage as the weather forecasts are the most up-to-date. The stochastic model is simplified by making ground holding decisions on groups of aircraft defined according to different characteristics. Alonso *et al.* [46] later presented a stochastic multi-stage 0-1 model and a robust algorithmic framework for ATFM with sector and airport capacity uncertainty based on the Bertsimas-Stock model [47]. The authors propose two alternative versions based on simple and full recourse, which are compared with the deterministic problem that considers capacity expectation. The integer solution is reported to be within 0.25% of the optimal relaxed solution in most of the cases, indicating a good mathematical formulation.

Nilim and El Ghaoui [48] developed a stochastic dynamic programming algorithm for ATFM where the evolution of the weather is modeled as a stationary Markov chain. Their results provide a dynamic routing strategy that minimizes the expected system delay. Performed simulation suggests that a significant improvement in delay can be obtained by using their approach. Mukherjee and Hansen [49] developed a linear dynamic stochastic optimization model for ATFM that considers uncertainty affecting both airport and sector capacity. The authors focused on the case of a single destination airport with a small number of arrival fixes subject to reduced capacity due to bad weather, with the main decisions assigning ground holding delays and local rerouting of inbound flights. Their experiments, conducted using data from the Dallas/Fort Worth International Airport, show that when weather impact is severe and persistent, substantial benefits in terms of delay savings are achieved by dynamically rerouting flights. Chang [50] analyzed stochastic programming approaches to ATFM under weather uncertainty in his PhD thesis. First, he proposes a two-stage SIP that indicates how aircraft should be sent toward a sector subject to weather uncertainty. However, due to the large number of weather scenarios, the model is intractable in practice. This issue is faced using a rolling horizon method that involves Lagrangian relaxation and the subgradient method. The two-stage model is then extended to the multi-stage version of the problem, which increases the number of possible weather realizations.

Andreatta *et al.* [51] formulated an aggregated stochastic model for the ATFM problem under airport capacity uncertainty. The authors consider the multi-airport capacity allocation problem to manage congestion phenomena in ATFM. Their model suggests how many flights should be delayed during each time period under consideration, rather than providing detailed information. The objective is to compute an optimal mix of arrivals and departures for a given network of airports that minimizes the total delay over all the airports during the periods of congestion. Computed solutions are consistent with the Collaborative Decision Making (CDM) procedure in ATFM, which is widely adopted in the U.S. and is of great interest in Europe. Computational results for a maximum of 12 airports with 60 scenarios are reported. The Value of the Stochastic Solution (VSS) is reported around 5.74% on average. The results for the instances indicate a tradeoff between a higher number of departures and a lower number of arrivals at both airports, compared to the deterministic solution. The first-stage stochastic solution is less restrictive, in terms of ground delay assignment, than the corresponding deterministic solution.

A. Agustín *et al.* [52] presented a framework for modeling multi-stage mixed 0-1 problems for ATFM with rerouting under uncertainty on airport and air sector capacities and flight demand. Five different objective functions are considered, and a scenario tree based scheme is used to represent the Deterministic Equivalent Model (DEM) of the stochastic mixed 0-1 program with full recourse. The nonanticipativity constraints that equate the common 0-1 and continuous variables from the same group of scenarios in each period are implicitly satisfied in the compact representation of the DEM. Computational experiments are reported for medium-scale instances. The authors report that the model is so tight that the branch-and-cut phase from a commercial MIP solver is not required for most instances.

Bertsimas and Gupta [53] presented the first application of robust and adaptive optimization to the ATFM problem. They consider the stochastic ATFM problem with uncertainty on airport and sector capacity induced by weather, incorporating both adaptability and robustness. An uncertainty set for robust optimization is constructed based on a small number of bad weather fronts moving across different parts of the U.S. National Airspace System (NAS). Tractable solution methodologies are proposed, and the equivalence of the robust problem to a new deterministic instance is proven. Computational experiments are reported on instances based on real data augmented with simulated weather fronts. Instances of the robust problem that consider a number of flights between 500 and 1,000 are solved in solution times that range between 6' and 78'. The authors highlight that the robust problem inherits the attractive properties of the deterministic problem, and that the price of robustness is typically small. In the following section, a new stochastic model for the ATFM problem that considers stochastic capacity availability at airports and air sectors is discussed.

3.2 Stochastic optimization model

A new two-stage stochastic integer programming model for the SATFM problem is now presented. This model considers a geographical area that includes the corresponding available airspace and a network of selected airports. Its objective is to minimize the cost of delays of flights scheduled over a specific time horizon at these airports. Both airport and sector capacity constraints are considered, taking the related uncertainty due to bad weather conditions into account. Information on possible capacity reductions can be derived from available weather forecasts. The model produces flight plans that are robust against the bad weather fronts that are expected on the considered airspace.

To formulate the proposed model, some assumptions are made, and a specific set of control options is considered. The rerouting option for flights is considered. However, for each flight only a “small" set of possible routes is considered. This is important for the computational experiments. Control options also include the possibility to assign both ground holding and airborne delay. However, airborne delay is assumed to be only assignable to a flight in the terminal area of its destination airport, i.e., the last sector in the flight route, therefore speed adjustment along the route is not considered. In view of this assumption, the sector flight time for each flight is considered constant and known in advance, except for the last sector before the destination airport, where airborne delay can take place. This assumption is motivated considering that the model is executed at the tactical phase, and including more operational decisions in the model at this phase is somewhat questionable. Furthermore, in most ATFM instances, it may not even be possible to capture such effect. Indeed, the time spent by a flight to cross a sector is usually at most 20 minutes. The speed variation of a flight is of the order of few percentage points. Supposing a 10% flight speed reduction, the flight time is equal to 22 minutes. With the time horizon divided into 10 or 15-minute time instants, it is not possible to detect or measure the delay due to operational decisions.

Two different sets of decisions that can be made on flights are proposed in the following, which result in two different formulations for the model. Such formulations are referred to in the following as *flight plan formulation* and *route formulation*. Flights are subdivided into three sets in both formulations, with the two formulations only differing on the definition of the second set of flights. The three sets of flights are defined as follows:

- **Set** $\mathcal{F}_1$. On upcoming flights, a final flight plan is defined. The departure time is fixed and cannot be delayed. The arrival time may be delayed by applying airborne delay to flights. This definition is valid for both formulations.

- **Set** $\mathcal{F}_2$. For the following flights, an initial flight plan is determined in the flight plan formulation. This flight plan is evaluated over future scenarios and both ground holding and airborne delays may be applied. For the route formulation, only the route is chosen for this set of flights, without defining a complete flight plan.

- **Set** $\mathcal{F}_3$. Flights that are far away into the future are only used to estimate the cost of their delays using their standard (most commonly used) route at each scenario realization. Delay decisions are not applied in practice, as they are only made for the sake of future cost estimation. This definition is valid for both formulations.

An example of use of this subdivision of flights is illustrated in Figure 3.1, where an eight-hour time horizon is considered. Flights departing in the first hour of this time horizon belong to $\mathcal{F}_1$. Flights departing in the following two hours belong to $\mathcal{F}_2$. Finally, flights departing in the final five hours of the time horizon belong to $\mathcal{F}_3$. A final decision is made on the departure time of upcoming flights, i.e., flights departing between 8:00 and 9:00, and an initial flight plan (or the preferred route, in case of use of the route formulation) is determined for flights taking place in the next two hours.

In the following, the two mathematical formulations proposed for the new model are illustrated. The notation, decision variables, objective function and constraints for the first and second-stage of the formulations are provided in §3.2.1 and 3.2.2, respectively.

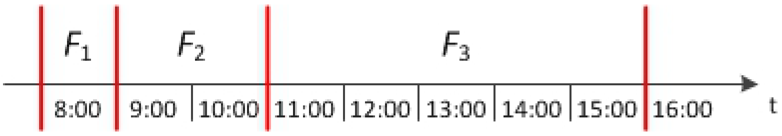

Figure 3.1: TIME HORIZON SUBDIVISION FOR DETERMINING FLIGHT CATEGORIES

3.2.1 First-stage formulations

Notation

The first-stage formulations use the following notation:

$\mathcal{A} \equiv$ set of airports, indexed by a

$\mathcal{S} \equiv$ set of sectors, indexed by s

$\mathcal{F}_1 \equiv$ set of flights for which a final decision is made on the departure time

$\mathcal{F}_2 \equiv$ set of flights for which an initial flight plan is computed

$\mathcal{F} \equiv$ set of flights, $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$, indexed by f

$\underline{\mathcal{F}} \equiv$ set of flights scheduled in the first-stage, $\underline{\mathcal{F}} = \mathcal{F}$ for the flight plan formulation, $\underline{\mathcal{F}} = \mathcal{F}_1$ for the route formulation

$R^f \equiv$ set of routes that can be flown by flight $f \in \mathcal{F}$, indexed by r

$g_f : R^f \to 2^{|S|} \equiv$ map that assigns each route the subset of sectors included in it

$orig_f \equiv$ airport of origin of flight f

$dest_f \equiv$ airport of destination of flight f

$\mathcal{T} \equiv$ set of discrete time periods

$\tau^f_{r,j} \equiv$ minimum flight time of flight $f \in \mathcal{F}$ to get from the departure airport to airport/sector $j \in \mathcal{S} \cup dest_f$. For sectors, the flight time is fixed

$ft_f \equiv$ minimum flight time of flight $f \in \mathcal{F}$ from origin to destination

$TI^f_a \equiv$ planned time of use of airport a by flight f

$TF^f_a \equiv$ last instant at which flight f may use airport a, considering the maximum ground holding and airborne delays allowed

$T^f_a \equiv$ set of discrete time periods at which flight f may use airport a following any route, $T^f_a = [TI^f_a, TF^f_a]$

$T'_j \subseteq T \equiv$ set of time instants at which capacity is deterministic for airport/sector j

$T''_j \subseteq T \equiv$ set of time instants at which capacity is stochastic for airport/sector j

$\mathcal{P} \equiv$ set of flights (f, f'), with $f, f' \in \underline{\mathcal{F}}$, that are connected, with corresponding turnaround time $l_{ff'}$

$\alpha_a^t, \beta_a^t, \gamma_a^t \equiv$ departure, arrival and total capacity available at airport a at time t. For $t \in T_a'$ capacity is deterministic, for $t \in T_a''$ the value represents the maximum available capacity

$\theta_s^t \equiv$ capacity available at sector s at time t. For $t \in T_s'$, capacity is deterministic. For $t \in T_s''$, it represents the maximum available capacity

$L_j^f \equiv$ set of airport/sectors preceding airport/sector j over any route r for flight f

$N_j^f \equiv$ set of airport/sectors following airport/sector j over any route r for flight f

$GD \equiv$ maximum ground holding delay that can be assigned to a single flight

$AD \equiv$ maximum airborne delay that can be assigned to a single flight

Decision variables

In the first-stage, a decision is made on the complete flight plan for flights in $\mathcal{F}_1$ in both formulations, and for flights in $\mathcal{F}_2$ for the flight plan formulation. A decision on the complete flight plan means that the solution provides departure and arrival times at airports for flights, as well as the route to follow. Since the flight time between sectors is fixed, the departure time and the chosen route define sector entry times too. The decision at the origin is made by variables $xo_r^f(t)$. At the destination airport, the arrival time is defined by variables $xd^f(t)$, considering that airborne delay may take place in the last sector of the flight route only. For the route formulation, only the route is chosen in the first-stage for flights in $\mathcal{F}_2$. The decision on the route is made by using the y_r^f variables, which are not defined for the flight plan formulation.

$$xo_r^f(t) = \begin{cases} 1, & \text{if flight } f \in \underline{\mathcal{F}} \text{ departs from airport } orig_f \text{ by time } t \text{ following route } r, \\ 0, & \text{otherwise.} \end{cases}$$

$$xd^f(t) = \begin{cases} 1, & \text{if flight } f \in \underline{\mathcal{F}} \text{ arrives at airport } dest_f \text{ by time } t, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_r^f = \begin{cases} 1, & \text{if flight } f \in \mathcal{F} \setminus \underline{\mathcal{F}} \text{ follows route } r, \\ 0, & \text{otherwise.} \end{cases}$$

Objective function

As is the case with most ATFM models from literature, the proposed model minimizes a function which is a combination of the costs of airborne delay – denoted with AH – and ground holding delay – denoted with GH. The objective function has the two following properties:

1. A unit of airborne delay is more expensive than a unit of ground holding delay;
2. Flight delays are assigned "fairly".

Fairness of delay assignment asks for delays to be evenly distributed over different flights when such a distribution is possible. Accumulating all the delay on a single flight is an "unfair" decision. Hence, the objective function is defined as a super-linear function of the tardiness of a flight, i.e., it is the sum over the set of all flights of $AH^{1+\epsilon_2}$ and $GH^{1+\epsilon_1}$. Defining $\epsilon_2 > \epsilon_1$ it is possible to guarantee property 1, as airborne delay will be more expensive than ground holding delay. If ϵ_1 and ϵ_2 are close to zero, i.e., the terms of the objective function are slightly super-linear, the following approximation can be made:

$$AH_f^{1+\epsilon_2}+GH_f^{1+\epsilon_1} = AH_f^{1+\epsilon_2}+GH_f^{1+\epsilon_1}+GH_f^{1+\epsilon_2}-GH_f^{1+\epsilon_2} \cong TD_f^{1+\epsilon_2}-(GH_f^{1+\epsilon_2}-GH_f^{1+\epsilon_1})$$

where $TD_f = AH_f + GH_f$ is the total delay of flight f.

As first described by Lulli and Odoni [54], using total delay helps to overcome the following complication. If airborne and ground holding delay costs are accounted for separately – as is the case in most of the models from the existing literature – there is no distinction between the two following solutions. The first delays only one flight by assigning it one unit of airborne delay and one unit of ground holding delay. The second delays two flights by assigning one unit of ground holding delay to the first and one unit of airborne delay to the second. By contrast, if total delay is used, the model will favor the latter alternative, which is a preferable solution as it is more fair.

Summarizing, the objective function is composed of two terms. The first term provides the cost of the total delay assigned to a flight. The second term accounts for the cost reduction obtained when a part of the total delay takes place on the ground, before taking off. Hence, for each flight $f \in \mathcal{F}$ and for each time period $t \in \mathcal{T}$, the following two cost coefficients are defined:

$$c_{td}^f(t) = (t-a_f)^{1+\epsilon_2} \equiv \text{total cost of delaying flight } f \text{ by } (t-a_f) \text{ units of time}$$

$$c_g^f(t) = (t-d_f)^{1+\epsilon_2} - (t-d_f)^{1+\epsilon_1} \equiv \text{cost reduction obtained by holding flight } f \text{ on the ground for } (t-d_f) \text{ units of time}$$

where $a_f = TI^f_{dest_f}$ and $d_f = TI^f_{orig_f}$ are the scheduled arrival and departure times of flight f, respectively. The objective function can now be formulated as follows:

$$Min \sum_{f\in\mathcal{F},t\in T^f_{dest_f}} c^f_{td}(t)\cdot\big(xd^f(t)-xd^f(t-1)\big)- \tag{3.1a}$$
$$\sum_{f\in\mathcal{F},r\in R^f,t\in T^f_{orig_f}} c^f_g(t)\cdot\big(xo^f_r(t)-xo^f_r(t-1)\big)+E[f(xo,xd,\tilde{\omega})]$$

Constraints
The objective function is optimized over the set of feasible solutions described by the following sets of constraints.

Capacity constraints

$$\sum_{f\in\mathcal{F},r\in R^f}\big(xo^f_r(t)-xo^f_r(t-1)\big)\le\alpha^t_a \quad \forall a\in\mathcal{A},t\in\mathcal{T} \tag{3.1b}$$
$$\sum_{f\in\mathcal{F}}\big(xd^f(t)-xd^f(t-1)\big)\le\beta^t_a \quad \forall a\in\mathcal{A},t\in\mathcal{T} \tag{3.1c}$$
$$\sum_{f\in\mathcal{F}}\Big(xd^f(t)-xd^f(t-1)+\sum_{r\in R^f}\big(xo^f_r(t)-xo^f_r(t-1)\big)\Big)\le\gamma^t_a \quad \forall a\in\mathcal{A},t\in\mathcal{T} \tag{3.1d}$$
$$\sum_{f\in\mathcal{F},r\in R^f:s\in g_f(r)}\big(xo^f_r(t-\tau^f_{r,s})-xo^f_r(t-\tau^f_{r,s}-1)\big)\le\theta^t_s \quad \forall s\in\mathcal{S},t\in\mathcal{T} \tag{3.1e}$$

Constraints (3.1b), (3.1c) and (3.1d) enforce the departure, arrival and total capacity limitations at airports, respectively. Sector capacity constraints are defined by (3.1e). The capacity of a sector $s\in\mathcal{S}$ is defined as the maximum number of entries in the sector per time period, according to the European definition of sector capacity. It is however possible to easily adapt the model to the U.S. definition of sector capacity, which considers the number of flights present in a sector at a given time period. Constraints (3.1f) provide the sector capacity definition for the U.S..

$$\sum_{f\in\mathcal{F},r\in R^f:s\in g_f(r)}\left(xo^f_r(t-\tau^f_{r,s})-\begin{cases} xo^f_r(t-\tau^f_{r,N^f_r(s)}) & \text{if } N^f_r(s)\neq dest_f\\ xd^f(t) & \text{otherwise}\end{cases}\right)\le\theta^t_s \quad \forall s\in\mathcal{S},t\in\mathcal{T} \tag{3.1f}$$

Route selection

$$\sum_{r \in R^f} y_r^f = 1 \quad \forall f \in \mathcal{F} \setminus \underline{\mathcal{F}} \tag{3.1g}$$

$$\sum_{r \in R^f} xo_r^f(TF^f_{orig_f}) = 1 \quad \forall f \in \underline{\mathcal{F}} \tag{3.1h}$$

$$\sum_{r \in R^f} xo_r^f(t - \tau^f_{r,dest_f}) \geq xd^f(t) \quad \forall f \in \underline{\mathcal{F}}, t \in T^f_{\mathcal{D}(f)} \tag{3.1i}$$

$$xd^f(t + ft_f + AD) \geq \sum_{r \in R^f} xo_r^f(t) \quad \forall f \in \underline{\mathcal{F}}, t \in T^f_{orig_f} \tag{3.1j}$$

$$xd^f(TF^f_{dest_f}) = 1 \quad \forall f \in \underline{\mathcal{F}} \tag{3.1k}$$

These constraints take care of the routes followed by flights. To ensure that exactly one route is selected for each flight, constraints (3.1g) take care of flights from set $\mathcal{F}_2$ for the route formulation, while constraints (3.1h) take care of flights from set $\mathcal{F}_1$ and for flights from set $\mathcal{F}_2$ for the flight plan formulation. Constraints (3.1g) are not present in the flight plan formulation. Constraints (3.1i) guarantee that the minimum flight time for the chosen route is respected. Constraints (3.1j) guarantee that the maximum airborne delay is not exceeded. Finally, constraints (3.1k) guarantee the arrival of a flight.

Connectivity

$$xd^f(t) \geq \sum_{r \in R^f} xo_r^{f'}(t + l_{ff'}) \qquad \forall (f, f') \in \mathcal{P}, t \in T^f_{dest_f} : t + l_{ff'} \in T^f_{orig_{f'}} \tag{3.1l}$$

Constraints (3.1l) guarantee that the turnaround time between connected flights is respected, by properly spacing the arrival and departure movements of two connected flights.

Decision variables definition

$$xo_r^f(t) \leq xo_r^f(t+1) \qquad \forall f \in \underline{\mathcal{F}}, r \in R^f, t \in T^f_{orig_f} \tag{3.1m}$$

$$xd^f(t) \leq xd^f(t+1) \qquad \forall f \in \underline{\mathcal{F}}, t \in T^f_{dest_f} \tag{3.1n}$$

$$xo_r^f(t), xd^f(t), y_r^f \in \{0, 1\} \qquad f \in \underline{\mathcal{F}}, r \in R^f, t \in T^f_{orig_f} \cup T^f_{dest_f} \tag{3.1o}$$

Constraints (3.1m) and (3.1n) guarantee that the xo_r^f and xd^f decision variables monotonically increase over time. Finally, constraints (3.1o) define decision variables as binary.

The proposed formulations are advantageous as the number of variables for each flight f is small. For example, consider a flight for which a complete flight plan should be defined. The number of variables needed for the departure movement is equal to the number of time instants at which the departure may take place, multiplied by the number of routes it may follow. If the considered flight has three alternative routes and a maximum ground holding delay of 90', the number of variables to define its departure is $6 \cdot 3 = 18$, considering a time horizon subdivided into 15' time instants. For the arrival, possible airborne delay in the last sector of a route has to be taken into account. However, it is not necessary to make a decision on the route, as all routes lead to the same airport, reducing the number of variables needed. If the maximum airborne delay for the considered flight is 30', the number of variables is equal to eight (six for ground holding delay, plus two for airborne delay). The total number of variables for the considered flight is equal to 21. Notice that the number of variables needed to represent the complete route of a flight is independent of the number of sectors it may fly through. This property is valid for the assumption of fixed flight times between sectors. Also, there is no need to formulate explicit constraints on the sequence of sectors a flight may fly through. This is due to the definition of the $xd_r^f(t)$ variables, which make a joint decision on the departure time and the route to be followed by a flight.

3.2.2 Second-stage formulations

Notation

The second-stage formulations present the following additional notation:

$\mathcal{F}_3 \equiv$ set of flights for which future delays are only evaluated on their standard route

$\overline{\mathcal{F}} \equiv$ set of flights that may be assigned ground holding delay, $\overline{\mathcal{F}} = \mathcal{F}_2 \cup \mathcal{F}_3$

$\mathcal{F}' \equiv$ set of all flights, $\mathcal{F}' = \mathcal{F} \cup \mathcal{F}_3$, indexed by f

$d_t^a \equiv$ amount of departure capacity reserved to flights of set $\mathcal{F}_1$ at airport a and time t in the first-stage

$c_t^s \equiv$ amount of capacity reserved to flights of set $\mathcal{F}_1$ at sector s and time t in the first-stage, with s not being the flights' terminal sector

$\Omega \equiv$ set of scenarios, indexed by ω

$$\overline{\alpha}_t^a(\omega), \overline{\beta}_t^a(\omega), \overline{\gamma}_t^a(\omega) \equiv \text{departure, arrival and total capacity available at airport } a \text{ at time } t \text{ under scenario } \omega$$
$$\overline{\theta}_t^s(\omega) \equiv \text{capacity of sector } s \text{ at time } t \text{ under scenario } \omega$$

Furthermore, notation from the first-stage is expanded to consider flight $f \in \mathcal{F}_3$ too. Notice that the route set for these new flights is formed by a single element, which represents the preferred route.

Decision variables
The decision variables used in the second-stage are the following:

$$zo_r^f(t) = \begin{cases} 1, & \text{if flight } f \in \overline{\mathcal{F}} \text{ departs by time } t \text{ following route } r \\ 0, & \text{otherwise.} \end{cases}$$
$$zd^f(t) = \begin{cases} 1, & \text{if flight } f \in \mathcal{F}' \text{ arrives at destination by time } t \\ 0, & \text{otherwise.} \end{cases}$$

These decision variables are of the same form as the $xo_r^f(t)$ and $xd^f(t)$ variables from the first-stage and determine when a flight departs/arrives from/at airport a, along with the chosen route. Notice that these decision variables are only defined at the destination airport for flights $f \in \mathcal{F}_1$, as such flights may only be assigned airborne delay. The route to be followed is fixed by the constraints, as it is either set in the first-stage for flights $f \in \mathcal{F}$, or it is the standard route for flights $f \in \mathcal{F}_3$.

Objective function
In the second-stage, flights belonging to set $\mathcal{F}_1$ can only be assigned additional airborne delay. Under a flight plan formulation, flights belonging to set $\mathcal{F}_2$ may receive additional ground holding or airborne delay, and they follow the route $r \in R^f$ chosen in the first-stage. Under a route formulation, they are only required to follow a specific route chosen in the first-stage, with delay being calculated over this route. Furthermore, the delay to assign to flights that belong to set $\mathcal{F}_3$ is estimated for each scenario realization $\omega \in \Omega$ considering their execution over their standard route r^f, i.e., the route they use the most frequently. The second-stage objective function is defined as follows:

$$
\begin{aligned}
f(xo, xd, \omega) = Min \sum_{f \in \mathcal{F}', t \in T} & c_{td}^f(t) \cdot \Big(zd^f(t) - zd^f(t-1) \Big) - \\
\sum_{f \in \overline{\mathcal{F}}, r \in R^f, t \in T} & c_g^f(t) \cdot \Big(zo_r^f(t) - zo_r^f(t-1) \Big) - \\
& (\mathcal{F}_1 \text{ ground delay cost} + \underline{\mathcal{F}} \text{ delay cost})
\end{aligned} \tag{3.2a}
$$

This objective function is the summation of the additional delay cost assigned to flights that belong to $\underline{\mathcal{F}}$ and the estimate of the cost of delays assigned to flights that belong to $\mathcal{F}' \setminus \underline{\mathcal{F}}$. Cost coefficients $c_{td}^f(t)$ and $c_g^f(t)$ are naturally extended to flights $f \in \mathcal{F}' \setminus \underline{\mathcal{F}}$. To only consider additional costs for flights $f \in \underline{\mathcal{F}}$, the cost of delay of their original schedule is subtracted from their total delay cost. Since the second addend does not consider flights in $\mathcal{F}_1$, it is necessary to separately subtract the ground delay cost for these flights. This cost can be retrieved from the first-stage objective value. This is also done to retrieve the delay cost assigned in the first-stage to all flights $f \in \underline{\mathcal{F}}$. Since they are both constant values, they do not influence the second-stage decision. It is however important to include them in the objective function to give the second-stage objective function its correct weight in the two-stage decision making process.

Constraints

The second-stage formulations have the following sets of constraints.

Capacity constraints

$$
\sum_{f \in \overline{\mathcal{F}}, r \in R^f} \big(zo_r^f(t) - zo_r^f(t-1) \big) \leq \begin{cases} \alpha_t^a - d_t^a & \forall a \in \mathcal{A}, t \in T'_a \\ \overline{\alpha}_t^a(\omega) & \forall a \in \mathcal{A}, t \in T''_a \end{cases} \tag{3.2b}
$$

$$
\sum_{f \in \mathcal{F}'} \big(zd^f(t) - zd^f(t-1) \big) \leq \begin{cases} \beta_t^a & \forall a \in \mathcal{A}, t \in T'_a \\ \overline{\beta}_t^a(\omega) & \forall a \in \mathcal{A}, t \in T''_a \end{cases} \tag{3.2c}
$$

$$
\sum_{f \in \overline{\mathcal{F}}, r \in R^f} \big(zo_r^f(t) - zo_r^f(t-1) \big) + \sum_{f \in \mathcal{F}'} \big(zd^f(t) - zd^f(t-1) \big) \leq \begin{cases} \gamma_t^a - d_t^a & \forall a \in \mathcal{A}, t \in T'_a \\ \overline{\gamma}_t^a(\omega) & \forall a \in \mathcal{A}, t \in T''_a \end{cases} \tag{3.2d}
$$

$$\sum_{f\in\overline{\mathcal{F}},r\in R^f:s\in g_f(r)} \left(zo_r^f(t-\tau_{r,s}^f)-zo_r^f(t-\tau_{r,s}^f-1)\right)\leq\begin{cases}\theta_t^s-c_t^s & \forall s\in\mathcal{S},t\in T_s'\\ \overline{\theta}_t^s(\omega) & \forall s\in\mathcal{S},t\in T_s''\end{cases} \tag{3.2e}$$

Constraints (3.2b) are the second-stage equivalent of constraints (3.1b), defined for time instants with deterministic and stochastic departure capacity. The first possible right-hand side term corresponds to the residual capacity available after scheduling flights $f\in\mathcal{F}_1$, while the second right-hand side term considers stochastic capacity only. Similarly, constraints (3.2c) address arrival capacity, being the second-stage version of constraints (3.1c), and constraints (3.2d) address total capacity, being the second-stage version of constraints (3.1d). Finally, constraints (3.2e) correspond to constraints (3.1e) from the first-stage, regulating sector capacity according to the European definition. Constraints (3.2f), discussed in the following, are the adaptation of the sector capacity constraints to the U.S. definition of sector capacity. Constraints operating in the stochastic time horizon are characterized by capacity availability that depends on the scenario realization $\omega\in\Omega$. Notice that the left-hand side considers all sectors for flights $f\in\overline{\mathcal{F}}$ and only the terminal sector for flights $f\in\mathcal{F}_1$. In fact, for latter set of flights, airborne delay may be applied in the last sector of the flight route in the second-stage, while the rest of the flight plan is fixed and cannot be changed.

$$\sum_{f\in\overline{\mathcal{F}},r\in R^f:s\in g_f(r)} \left(zo_r^f(t-\tau_{r,s}^f)-\begin{cases}zo_r^f(t-\tau_{r,N_r^f(s)}^f) & \text{if } N_r^f(s)\neq dest_f\\ zd^f(t) & \text{otherwise}\end{cases}\right)+$$
$$\sum_{f\in\mathcal{F}_1:r\in R^f:N_r^f(s)=dest_f} \left(xo_r^f(t-\tau_{r,s}^f)-zd^f(t)\right)\leq\begin{cases}\theta_t^s-c_t^s & \forall s\in\mathcal{S},t\in T_s'\\ \overline{\theta}_t^s(\omega) & \forall s\in\mathcal{S},t\in T_s''\end{cases} \tag{3.2f}$$

First/second-stage link

$$zo_r^f(t)\leq xo_r^f(t)\quad\forall f\in\underline{\mathcal{F}}\setminus\mathcal{F}_1,r\in R^f,t\in T_{orig_f}^f \tag{3.2g}$$
$$zd^f(t)\leq xd^f(t)\quad\forall f\in\underline{\mathcal{F}},t\in T_{dest_f}^f \tag{3.2h}$$

Constraints (3.2g) and (3.2h) link first and second-stage variables, ensuring that flights $f\in\mathcal{F}$ can only be assigned delay (possibly null) in the second-stage. Ground holding delay may be assigned to flights in $\mathcal{F}_2$ only, while airborne delay can be applied to all flights in $\mathcal{F}$. Flights $f\in\mathcal{F}_3$ are not considered in the first-stage so there is no link with this stage.

Route selection

$$zo_r^f(TF_{orig_f}^f) = y_r^f \qquad \forall f \in \mathcal{F} \setminus \underline{\mathcal{F}} \tag{3.2i}$$

$$\sum_{r \in R^f} zo_r^f(TF_{orig_f}^f) = 1 \quad \forall f \in \overline{\mathcal{F}} \tag{3.2j}$$

$$zd^f(t) \leq \sum_{r \in R^f} \begin{cases} xo_r^f(t - \tau_{r,dest_f}^f) & \forall f \in \mathcal{F}_1, t \in T_{dest_f}^f \\ zo_r^f(t - \tau_{r,dest_f}^f) & \forall f \in \overline{\mathcal{F}}, t \in T_{dest_f}^f \end{cases} \tag{3.2k}$$

$$zd^f(t + ft_f + AD) \geq \sum_{r \in R^f} \begin{cases} xo_r^f(t) & \forall f \in \mathcal{F}_1, t \in T_{orig_f}^f \\ zo_r^f(t) & \forall f \in \overline{\mathcal{F}}, t \in T_{orig_f}^f \end{cases} \tag{3.2l}$$

$$zd^f(TF_{dest_f}^f) = 1 \quad \forall f \in \mathcal{F}' \tag{3.2m}$$

Constraints (3.2i) take advantage of constraints (3.1g) from the first-stage, which define the route followed by a flight $f \in \mathcal{F}_2$ for the route formulation. These constraints are not present in the flight plan formulation. With the route formulation, routes for these flights are set in the first-stage, so variables corresponding to routes that are not followed can be fixed to 0. Constraints (3.2j) correspond to constraints (3.1h) from the first-stage, which ensure that each flight departs and follows exactly one route. Constraints (3.2k) impose the minimum duration of a flight, similarly to first-stage constraints (3.1i). Constraints (3.2l) are the second-stage version of constraints (3.1j), and ensure that the maximum airborne delay is respected. Finally, constraints (3.2m) correspond to constraints (3.1k) from the first-stage, that ensure the arrival of flights at destination.

Connectivity

$$zd^f(t) \geq \sum_{r \in R^f} zo_r^{f'}(t + l_{ff'}) \quad \forall (f, f') \in \mathcal{P}, t \in \mathcal{T} \tag{3.2n}$$

Constraints (3.2n) correspond to first-stage constraints (3.1l). They ensure that the turnaround time is respected.

Decision variables definition

$$zo_r^f(t) \leq zo_r^f(t+1) \qquad \forall f \in \overline{\mathcal{F}}, r \in R^f, t \in T_{orig_f}^f \tag{3.2o}$$

$$zd^f(t) \leq zd^f(t+1) \qquad \forall f \in \mathcal{F}', t \in T_{dest_f}^f \tag{3.2p}$$

$$zo_r^f(t), zd^f(t) \in \{0,1\} \qquad \forall f \in \mathcal{F}', r \in R^f, t \in T_{orig_f}^f \cup T_{dest_f}^f \tag{3.2q}$$

Constraints (3.2o) and (3.2p) link variables that are consecutive in time, defining them as monotonically increasing. Finally, constraints (3.2q) define all variables as binary.

3.3 Progressive Binary Heuristic

To efficiently solve instances of the SATFM problem, an ad-hoc heuristic, called Progressive Binary Heuristic (PBH), is developed and used. It is a stage-wise heuristic based on the specific characteristics of the mathematical formulations described in §3.2. The solution of the DEP (see §1.4) with relaxed second-stage variables is binary in most analyzed instances, as discussed in the following section. This relaxed problem is easier to solve than the DEP, since all subproblems are linear and not binary programs. When its solution is binary, no further action needs to be taken, as an optimal binary solution is computed. When the solution is not binary, the computational experience reports a limited number of second-stage variables with non-binary solution. The PBH deals with this case, making second-stage variables converge to a binary solution, providing a solution that is (close to) optimal in short computation times.

The PBH method can be applied to any problem with the discussed characteristics. It is illustrated by the flowchart in Figure 3.2. The algorithm starts with reading the DEP with relaxed second-stage variables of the instance to solve. This problem is named RP. Also, the iteration counter i is initialized to 1, see step (1) in the flowchart. In step (2), RP is solved to optimality, providing a first-stage optimal solution (x^1, y^1) and a second-stage optimal solution $z^1(\omega)$ for each scenario $\omega \in \Omega$. These solutions are associated with the first and second-stage objective values v'_{RP} and v''_{RP}, respectively. In step (3), the problem's lower bound LB is set to the sum of the first and second-stage objective values. Also, an additional parameter v'' is set to the second-stage objective value. Step (4) checks whether all subproblems have a binary solution. If this is true, the PBH loop stops and the incumbent solution is declared either as optimal – see step (9a) – or ϵ-optimal – see step (9b). This depends on the comparison of the lower bound LB with the incumbent objective value $v'_{RP} + v''$, see step (8). Alternatively, the heuristic aims at making subproblems with some variable having a non-binary solution value converge to a binary solution. This is achieved by adding binary constraints to all binary variables with non-integer solution, see step (5). Since optimization solvers typically do not allow to change the type of a variable after their declaration, binary constraints are added for each second-stage variable $z_j(\omega)$ with non-binary solution with two operations. First, a new binary variable $\overline{z}_j(\omega)$ is added to the subproblem ω. Then, a constraint $z_j(\omega) - \overline{z}_j(\omega) = 0$

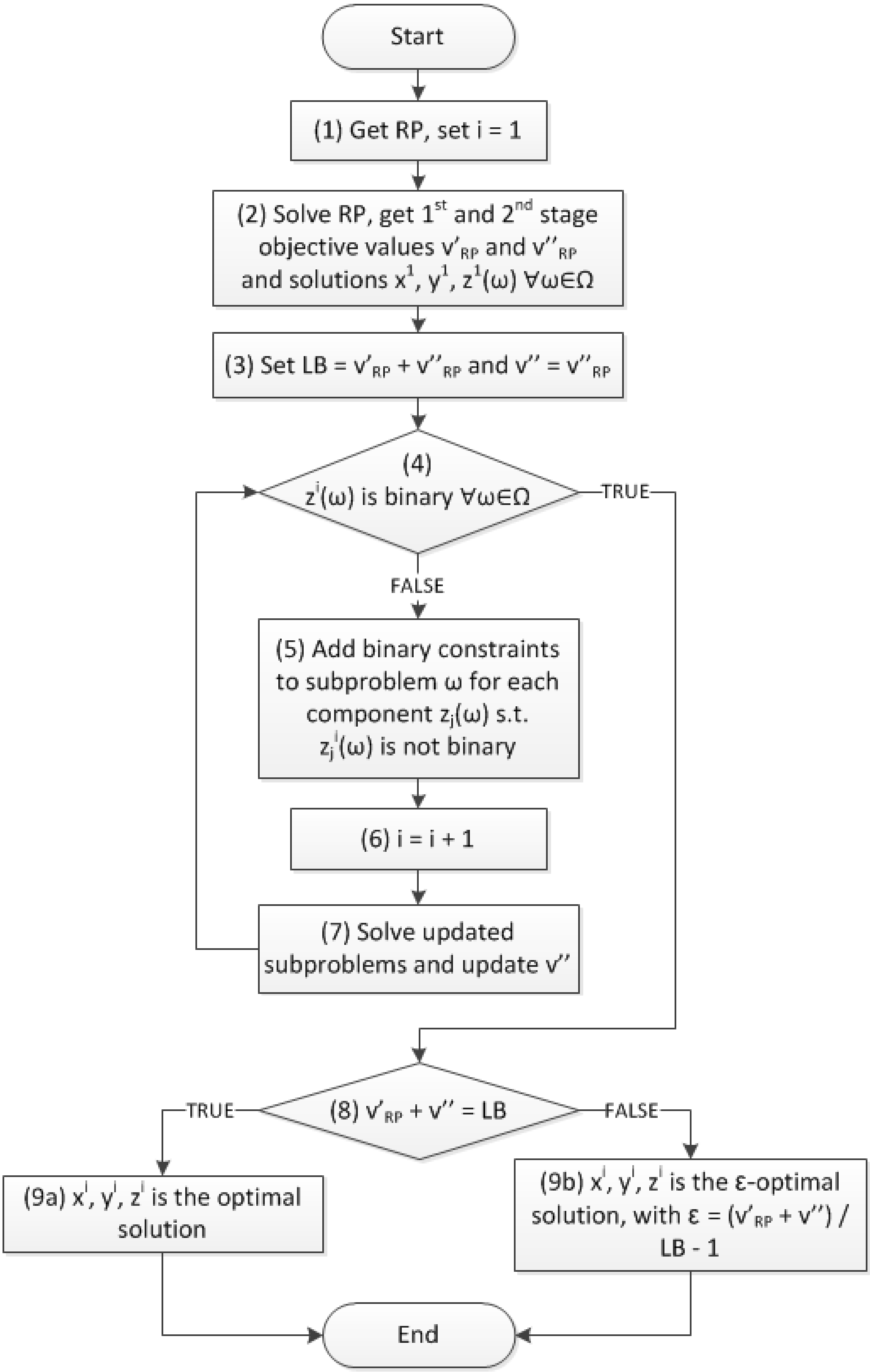

Figure 3.2: Progressive Binary Heuristic

to make the linear variable equal to the binary variable is added to the same subproblem. Next, the iteration counter i is updated in step (6). Finally, the iteration terminates solving the updated subproblems. The second-stage objective value v'' is updated if the newly solved subproblems provide a higher objective value compared with the previous solve. These operations are performed in step (7). The PBH loop then iterates, going back to step (4), to check that all subproblems have binary solutions.

Notice that step (5) of the heuristic, which adds binary constraints to subproblems, does not guarantee that the next subproblem solve will provide a binary solution. In fact, second-stage variables previously having a non-binary solution value will now change their value to either 0 or 1, and this may influence the value of other variables. For this reason, the heuristic may have to perform multiple iterations to converge to a binary solution, *progressively* adding binary constraints, hence the name of the heuristic.

Other similar approaches may be proposed. For example, it may be possible to only add a single binary constraint per iteration for each scenario, instead of adding constraints for all second-stage variables with a non-binary solution. This may provide a binary solution solving a problem that has fewer binary variables, which is easier to compute. For example, if two variables z_a and z_b that have a non-binary solution value are linked together by constraints such that $z_a + z_b = 1$, enforcing binary restrictions on one variable also enforces binary restrictions on the other. However, this approach may require a larger number of iterations. If ten unlinked second-stage variables have a non-binary solution, ten iterations may be required, solving the scenario subproblem ten times, instead of adding 10 variables and constraints and performing only one new subproblem solve. Therefore, this approach may be very inefficient in practice. Another approach may consist in enforcing binary constraints on all second-stage variables belonging to a subproblem with some non-binary solution value. This guarantees a binary solution at the following iteration, at the cost of solving a much more difficult problem. The proposed approach is a good compromise between these two extremes, as it solves subproblems with a limited number of binary variables and it avoids iterating too many times.

Finally, notice that the solution provided by the PBH is not guaranteed to be optimal. In fact, the heuristic only adds constraints to the second-stage subproblems, without affecting the first-stage. This means that the first-stage decisions are made only considering the relaxed subproblems, and the PBH provides the exact second-stage solution for each scenario, given the selected first-stage solution. In the following section, the PBH is used to solve different sets of instances of the SATFM problem, and its results are compared with those from the execution of the DEP.

3.4 SATFM Computational Results

The mathematical model for the SATFM problem was tested on four sets of realistically generated instances. The European airspace, i.e., the subdivision into air sectors, was simulated by defining a grid with squared cells with sides of approximately 100km. Airports and flights were then appropriately placed in this grid to simulate the real system, generating flight data realistically by considering real airport schedules.

Each instance set contains four different instances, which involve the same flights in the four sets. Instances involving the same flights consider different simulated weather conditions in each set. This difference is chosen to evaluate the effect of stochastic models on instances characterized by different types of stochastic information. For each instance, both the flight plan and route formulations were tested, resulting in eight different tests per instance set. The width of the time horizons to define the sets of flights $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$, as well as the total number of flights considered in each instance, is shown in Table 3.1. These figures apply to all instance sets. The airports considered in each instance are some of the busiest airports from Central/Western Europe, and they are the following:

- *4central* and *4central8h*: London Heathrow (EGLL), Paris - Charles de Gaulle (LFPG), Amsterdam - Schiphol (EHAM), Frankfurt (EDDF)
- *6central* and *6central8h*: London Heathrow (EGLL), Paris - Charles de Gaulle (LFPG), Amsterdam - Schiphol (EHAM), Frankfurt (EDDF), Dusseldorf (EDDL), Munich (EDDM)

Bad weather generation is crucial in the definition of the different sets of instances. For each instance, nine different bad weather scenarios are generated in the time horizon of $\mathcal{F}_2$ and $\mathcal{F}_3$. A basic capacity reduction is applied, and scenarios consider the possibility for bad weather to either enforce the basic capacity reduction, a heavier capacity reduction, or a lighter capacity reduction. This generates three different scenarios. Each scenario is then branched again considering bad weather staying the same, getting heavier or getting

Table 3.1: Description of the instances

Name	**$\mathcal{F}_1$ time horizon (h)**	**$\mathcal{F}_2$ time horizon (h)**	**$\mathcal{F}_3$ time horizon (h)**	**# Flights**
4central	1	1	2	1234
4central8h	2	2	4	2381
6central	1	1	2	1620
6central8h	2	2	4	3163

lighter in the middle of the stochastic time horizon, thus producing the final nine scenarios. Parameters on the width of the bad weather fronts and on the increase/decrease probability and percentage differ for each instance set. The different characteristics of each of the four sets are illustrated in Table 3.2. Here, column 1 (Instance set) provides the name of the instance set. Column 2 (Basic red.) shows the basic capacity reduction applied to sector and airport capacity constraints affected by the bad weather front. Columns 3 (Prob. better weather), 4 (Prob. same weather) and 5 (Prob. worse weather) show the probability of bad weather getting lighter, staying the same or getting heavier, respectively. These probabilities apply both at the beginning of the stochastic time horizon and in the middle of the time horizon, when bad weather conditions might change. The percentage of capacity increase or decrease when the bad weather front gets lighter or heavier is shown in columns 6 (Cap. incr. with better weather) and 7 (Cap. decr. with worse weather), respectively. Finally, column 8 (Front location) describes the behavior of the bad weather front. For sets 1 to 3, the fronts affect a squared area with a side of approximately 500km, starting with a North-Western position in London, and then moving East at approximately 100km/h. For set 4, on the other hand, bad weather affects the whole network for the whole time horizon. These parameters reflect the following behaviors of the bad weather fronts:

- *Set 1*: The bad weather front causes a mild capacity reduction, which does not have a very high probability to change over time. Furthermore, bad weather is not expected to get much heavier.
- *Set 2*: Same characteristics as set 1, with a stronger bad weather front that cuts off half of the capacity resources in the affected area.
- *Set 3*: The evolution of bad weather is subject to higher uncertainty, i.e., the probability of having either worse or better weather is higher compared with parameters from sets 1 and 2. Also, capacity increases or decreases in the respective scenarios by a higher percentage, making scenarios more diverse.
- *Set 4*: Bad weather is not very heavy, as the basic capacity reduction is lighter than in the other sets. Capacity variations over scenarios are not as heavy as in set 3, however scenarios have the same probability distribution as set 3. In this set, the key factor is the geographic distribution of the bad weather front, as it involves the whole network for the whole stochastic time horizon.

Table 3.2: Bad weather characteristics by instance set

Instance set	**Basic red.**	**Prob. better weather**	**Prob. same weather**	**Prob. worse weather**	**Cap. incr. with better weather**	**Cap. decr. with worse weather**	**Front location**
Set 1	30%	20%	60%	20%	20%	10%	EGLL → East
Set 2	50%	20%	60%	20%	20%	10%	EGLL → East
Set 3	30%	30%	40%	30%	30%	30%	EGLL → East
Set 4	20%	30%	40%	30%	20%	20%	Whole network

The different generated bad weather fronts affect the difficulty to solve problem instances. In particular, to avoid infeasibility on some instances, it is necessary to allow for larger ground holding and airborne delays. Since the maximum allowed amount of both kinds of delay influences the number of variables in the model, the need for larger delays corresponds to solving larger instances. The dimensions of all considered instances are shown in Table 3.3 for sets 1 and 3, whose corresponding instances share the same dimensions. Then, for instances from sets 2 and 4, the size of each instance is provided by Tables 3.4 and 3.5, respectively. The three tables have the following structure. Column 1 (Instance) provides the name of the instance. Column 2 (Form.) specifies the formulation used to create the instance, with “FP” indicating the flight plan formulation and “R” indicating the route formulation. Column 3 (Max GH/A Delay) indicates the maximum ground holding and airborne delay that can be assigned to a single flight. Notice that the time horizon is subdivided into 15’ wide time instants. Finally, columns 4 (1^{st} Stage Columns), 5 (1^{st} Stage Rows), 6 (2^{nd} Stage Columns) and 7 (2^{nd} Stage Rows) show the size of the first and second-stage problems. All instances have nine second-stage problems, defined by the different scenarios generated. Notice that the number of variables is also influenced by the number of routes considered in the rerouting decisions. For the generated instances, at most three alternative routes are considered, which include the shortest path on the airspace grid and – at most – two close alternative routes.

Computational experiments were performed on a computer with Intel Core i7 CPU at 2.20GHz and 8GB of RAM. Computation times to solve each instance from the four sets using both the PBH and the DEP are reported in Tables 3.6-3.9. The PBH was implemented using the CPLEX 12.1 Callable Libraries [40] to solve optimization problems. The DEP was solved with CPLEX 12.4. The structure of Tables 3.6-3.9 is the following. Column 1 (Instance) provides the name of the instance. Column 2 (Form.) indicates the formulation used. Columns 3 (PBH Solution) and 4 (PBH Time (s)) show

Table 3.3: Size of instances from sets 1 and 3

Instance	Form.	Max GH/A Delay	1^{st} Stage Columns	1^{st} Stage Rows	2^{nd} Stage Columns	2^{nd} Stage Rows
4central	FP	4	13,732	17,442	17,636	34,019
4central	R	4	8,269	10,384	17,636	29,704
4central8h	FP	4	26,301	34,894	34,459	65,263
4central8h	R	4	15,191	18,984	34,459	56,497
6central	FP	4	17,494	22,063	23,285	44,601
6central	R	4	10,046	12,426	23,285	38,753
6central8h	FP	4	33,805	44,585	45,482	85,643
6central8h	R	4	19,354	23,915	45,482	74,308

Table 3.4: Size of instances from set 2

Instance	Form.	Max GH/A Delay	1^{st} Stage Columns	1^{st} Stage Rows	2^{nd} Stage Columns	2^{nd} Stage Rows
4central	FP	6	19,484	24,547	25,184	48,172
4central	R	6	11,433	14,313	25,184	41,843
4central8h	FP	8	48,330	63,528	63,931	120,693
4central8h	R	8	26,695	33,334	63,931	103,747
6central	FP	6	24,828	31,143	33,247	63,272
6central	R	6	13,856	17,151	33,247	54,700
6central8h	FP	8	62,145	81,509	84,398	158,550
6central8h	R	8	34,022	42,281	84,398	136,659

the optimal objective value and the solution time in seconds obtained by using the PBH. The optimality gap is not reported as it is always less than 0.1%. Column 5 (RP Sol. Nonbins.) shows the number of non-binary values in the solution of the DEP with relaxed second-stage. When this number is equal to 0, the PBH only solves the DEP with relaxed second-stage. Else, the PBH loop is executed to add binary constraints. Columns 6 (DEP Solution) and 7 (DEP Time (s)) show the optimal objective value and the computation time in seconds for solving the DEP with CPLEX. All problems were solved to optimality.

Table 3.5: Size of instances from set 4

Instance	Form.	Max GH/A Delay	1^{st} Stage Columns	1^{st} Stage Rows	2^{nd} Stage Columns	2^{nd} Stage Rows
4central	FP	4	13,732	17,442	17,636	34,019
4central	R	4	8,269	10,384	17,636	29,705
4central8h	FP	6	37,315	49,276	49,195	93,101
4central8h	R	6	20,943	26,183	49,195	80,245
6central	FP	4	17,494	22,063	23,285	44,601
6central	R	4	10,046	12,426	23,285	38,753
6central8h	FP	6	47,975	63,110	64,940	122,301
6central8h	R	6	26,688	33,124	64,940	105,688

Table 3.6: Instance set 1 - Solution times

Instance	Form.	PBH Solution	PBH Time (s)	RP Sol. Nonbins.	DEP Solution	DEP Time (s)	Time saved (s)
4central	FP	392.82	49.3	0	392.82	90.1	40.8
4central	R	392.82	20.6	4	392.82	17.1	-3.5
4central8h	FP	862.00	216.2	0	862.00	434.1	217.8
4central8h	R	862.00	55.6	0	862.00	183.2	127.6
6central	FP	437.28	73.3	0	437.28	150.6	77.3
6central	R	436.00	24.7	0	436.00	25.5	0.8
6central8h	FP	1121.56	237.8	0	1121.56	639.4	401.6
6central8h	R	1121.56	81.3	0	1121.56	81.4	0.1

Table 3.7: Instance set 2 - Solution times

Instance	Form.	PBH Solution	PBH Time (s)	RP Sol. Nonbins.	DEP Solution	DEP Time (s)	Time saved (s)
4central	FP	830.38	365.5	20	830.38	393.0	27.5
4central	R	829.95	49.1	0	829.95	45.1	-4.0
4central8h	FP	2048.88	1585.3	20	2048.88	1939.7	354.4
4central8h	R	2048.87	412.4	64	2048.87	807.0	394.6
6central	FP	875.07	513.6	46	875.07	656.8	143.2
6central	R	873.82	105.3	4	873.82	210.8	105.5
6central8h	FP	2542.93	1855.4	4	2542.93	2638.4	783.0
6central8h	R	2542.83	378.9	0	2542.83	1374.0	995.1

Finally, column 8 (Time saved (s)) shows the amount of time – expressed in seconds – saved by executing the PBH compared with the execution of the DEP.

In these tables, the number of non-binary solution values for all instances is very limited, as required by the PBH. Indeed, the maximum number of second-stage variables with a non-binary solution is 64 for instance *4central8h* with route formulation from set 2. Each second-stage problem in this instance has 63,931 variables, therefore only one out of every 8,990 second-stage variables does not have a binary value in the solution of

Table 3.8: Instance set 3 - Solution times

Instance	Form.	PBH Solution	PBH Time (s)	RP Sol. Nonbins.	DEP Solution	DEP Time (s)	Time saved (s)
4central	FP	422.97	54.3	0	422.97	121.3	67.0
4central	R	422.03	19.9	0	422.03	65.1	45.2
4central8h	FP	955.33	176.8	0	955.33	482.6	305.8
4central8h	R	955.33	63.6	0	955.33	220.2	156.6
6central	FP	466.60	89.0	4	466.60	91.7	2.7
6central	R	464.91	26.9	0	464.91	128.6	101.7
6central8h	FP	1225.84	362.2	0	1225.84	768.4	406.2
6central8h	R	1225.54	106.8	0	1225.54	103.1	-3.7

Table 3.9: INSTANCE SET 4 - SOLUTION TIMES

Instance	**Form.**	**PBH Solution**	**PBH Time (s)**	**RP Sol. Nonbins.**	**DEP Solution**	**DEP Time (s)**	**Time saved (s)**
4central	FP	252.98	66.2	36	252.98	33.5	-32.7
4central	R	252.97	39.1	22	252.97	49.8	10.7
4central8h	FP	835.86	498.2	0	835.86	988.7	490.5
4central8h	R	835.86	116.1	0	835.86	102.3	-13.8
6central	FP	386.58	62.1	0	386.58	60.0	-2.1
6central	R	386.57	27.3	0	386.57	24.0	-3.3
6central8h	FP	1134.00	642.2	0	1134.00	1284.0	641.8
6central8h	R	1133.80	192.5	0	1133.80	130.7	-61.8

the DEP with relaxed second-stage. The computational advantage of using the PBH on the considered instances is evident from the results reported in the tables. In just one out of the 32 considered cases the PBH takes one minute more than the DEP to solve the problem. Furthermore, for all cases where the PBH requires a longer computation time compared with the DEP, its computation times are always below 200". For more difficult instances, the PBH provides considerable savings in runtime. For example, consider the route formulation *6central8h* instance from set 2. The PBH is able to solve this instance in 6'18", compared with a runtime of 22'54" of the DEP. On average, the PBH allows to save 1'47" for set 1, 5'49" for set 2, 2'15" for set 3 and 2'08" for set 4, providing results within a 0.01% optimality gap. This confirms the good structure of the proposed formulations and the usefulness of the PBH.

Finally, to evaluate the practical advantage of using a stochastic model to solve the ATFM problem, the Value of the Stochastic Solution (VSS) is analyzed. The VSS represents the value provided by considering stochastic information. Without considering stochastic information, the decision maker can simply use the expected value of the stochastic data, substituting $\tilde{\omega}$ with $E[\tilde{\omega}]$ in the recourse function. A problem where this substitution is made is called Expected Value (EV) problem. The first-stage decisions made considering the EV problem can then be evaluated by implementing them in the two-stage program, which allows to get the expected result of the EV problem, denoted with EEV. The difference between the EEV and the optimal objective value of the stochastic problem is the VSS. The larger the VSS, the greater the benefit of using a stochastic model to solve the problem.

The study of the VSS for the different instance sets is presented in Tables 3.10-3.13. These tables have the following structure. Columns 1 (Instance) and 2 (Form.) provide the name of the instance and the formulation used, respectively. Column 3

Table 3.10: Instance set 1 - Value of the Stochastic Solution

Instance	Form.	DEP Sol.	EEV Sol.	VSS	VSS Pct.
4central	FP	392.82	397.23	4.41	1.1%
4central	R	392.82	394.26	1.44	0.4%
4central8h	FP	862.00	866.81	4.81	0.6%
4central8h	R	862.00	863.24	1.24	0.1%
6central	FP	437.28	443.99	6.71	1.5%
6central	R	437.00	438.04	1.04	0.2%
6central8h	FP	1121.55	1148.53	26.98	2.4%
6central8h	R	1121.55	1121.95	0.40	0.0%

Table 3.11: Instance set 2 - Value of the Stochastic Solution

Instance	Form.	DEP Sol.	EEV Sol.	VSS	VSS Pct.
4central	FP	830.38	838.87	8.49	1.0%
4central	R	829.95	830.75	9.80	0.1%
4central8h	FP	2048.88	2061.90	13.02	0.6%
4central8h	R	2048.87	2050.89	2.02	0.1%
6central	FP	875.07	880.45	5.38	0.6%
6central	R	873.82	876.27	2.45	0.3%
6central8h	FP	2542.93	2549.00	6.07	0.2%
6central8h	R	2542.83	2544.71	1.88	0.1%

(DEP Sol.) provides the optimal objective value obtained by solving each instance using the stochastic model. This is the best objective value that can be achieved for each instance. Column 4 (EEV Sol.), on the other hand, provides the EEV, which cannot by definition be lower than the DEP solution. The difference between the EEV and the DEP solution is reported in column 5 and it is the VSS. For a better understanding of its value, the VSS is reported in percentage terms in column 6.

Results report a limited VSS for instances from sets 1 to 3. On average, it is equal to 1.5%, 0.5% and 1.8% for instances formulated with the flight plan formulation for the first 3 sets. Using the route formulation, the VSS decreases to 0.1%, 0.1% and 0.3% on the 3

Table 3.12: Instance set 3 - Value of the Stochastic Solution

Instance	Form.	DEP Sol.	EEV Sol.	VSS	VSS Pct.
4central	FP	422.97	430.01	7.04	1.7%
4central	R	422.03	426.63	4.60	1.1%
4central8h	FP	955.33	970.72	15.39	1.6%
4central8h	R	955.33	957.01	1.68	0.2%
6central	FP	466.60	472.28	5.68	1.2%
6central	R	464.91	465.56	0.65	0.1%
6central8h	FP	1225.84	1254.20	28.36	2.3%
6central8h	R	1225.54	1226.73	1.19	0.1%

Table 3.13: INSTANCE SET 4 - VALUE OF THE STOCHASTIC SOLUTION

Instance	Form.	DEP Sol.	EEV Sol.	VSS	VSS Pct.
4central	FP	252.98	258.63	5.65	2.2%
4central	R	252.97	253.76	0.79	0.3%
4central8h	FP	835.86	958.26	122.40	14.6%
4central8h	R	835.86	838.05	2.19	0.3%
6central	FP	386.58	401.78	15.20	3.9%
6central	R	386.57	388.97	2.40	0.6%
6central8h	FP	1134.00	1214.23	80.23	7.1%
6central8h	R	1133.80	1135.97	2.17	0.2%

instance sets. Set 4 reports a much higher VSS value for the flight plan formulation, which is 8.6% on average. The route formulation, on the other hand, is still limited to 0.3%. These figures suggest that the proposed stochastic formulations provide little benefit when weather phenomena are local. On the other hand, the proposed stochastic model with flight plan formulation provides relevant benefit when bad weather phenomena are spread throughout the whole network. The route formulation, which makes limited decisions on flights from set $\mathcal{F}_2$ compared with the flight plan formulation, does not provide airlines with benefits that may justify the use of a stochastic model. The longer computation time needed by the flight plan formulation is therefore justified by the added value to the stochastic solution.

Using the study from the University of Westminster for EUROCONTROL [7] that estimates the average cost of one minute of ATFM delay to 83€, it is possible to provide an approximate upper bound estimate of the monetary savings that may be achieved using the stochastic model. The estimate is an upper bound of the savings due to the super-linear cost coefficients used in the definition of the objective function to favor ground holding over airborne delay and to implement fairness among flights. The multiplication of the VSS by the number of minutes within a single time period – which is equal to 15 – and by the 83€ per-minute ATFM delay cost estimate provides this approximate estimate of the monetary savings. For example, for flight plan instance *4central8h* from set 4, whose VSS is equal to 14.6%, the upper bound estimate of the savings amounts to 152,384€, corresponding to 64€ per flight.

3.5 SATFM Conclusions

Most research works on the ATFM problem focus on solving it using deterministic models, i.e., not considering its stochastic aspects. Stochastic models are more difficult to solve in nature than their corresponding deterministic versions, therefore some simplifications are usually adopted to make stochastic models solvable within computation times that are compatible with the practical needs of the application. The stochastic ATFM model presented considers uncertainty on capacity availability, both at airports and air sectors, assuming two simplifications. First, airborne delay may only take place in a flight's terminal airspace, i.e., the last sector crossed by a flight before landing. Second, rerouting is performed considering a limited number of routes.

Two different formulations are proposed, which make different decisions on flights. Also, taking advantage of the structure of the mathematical formulations proposed, an ad-hoc heuristic method is developed to improve the computation times of the problem instances. Four different sets of instances are analyzed, each characterized by simulated weather phenomena of different nature. Experiments on these instances show that the proposed heuristic, called Progressive Binary Heuristic, reduces the computation times provided by the Deterministic Equivalent Problem by over 40% on average. The Value of the Stochastic Solution is also analyzed to determine the benefit that airlines can obtain by using the proposed stochastic model to solve the ATFM problem. VSS figures indicate that the route formulation does not provide a tangible benefit, while the flight plan formulation provides airlines with a relevant delay cost reduction when bad weather affects the whole network. In this case, the VSS indicates that airlines may have reduced delay costs by up to 14% using the new stochastic model with flight plan formulation on a considered instance. The proposed stochastic programming model should therefore be used when uncertainty caused by bad weather is a factor that influences a wide portion of the managed airspace.

Future developments of this work include the development of a new algorithm to improve the presented computation times that can guarantee optimality for all instances. An approach based on Fenchel cutting planes [55] is currently being studied. Furthermore, experiments on additional instances involving wide bad weather fronts will be conducted to provide a more detailed analysis of the results reported in §3.4.

In the following section, the focus switches to a different type of uncertainty at the tactical decision phase. This is the implicit uncertainty on the time of departure of flights, and it is faced using deterministic optimization models.

3.6 The Air Traffic Flow Management Problem with Time Windows

The ATFM problem can be faced in different ways, involving both deterministic and stochastic optimization models. These models typically subdivide the time horizon into time instants of equal size. The operator of a flight is expected to adhere to its flight plan as precisely as possible. Specifying the execution of an action within a time instant corresponds in practice to the execution of said action within a period of time of predefined size. Some adjustments are usually possible. However, there are certain flights which must be operated in strict accordance with the approved flight plan, since even a small delay assigned to them may have a large downstream effect on the performance of the ATC system, leading to a degradation of the system itself. These flights are referred to as "critical flights" in the following. For these critical flights, there is no slack time in handling their operations and only a limited number of recovery options are generally available. Inversely, as some portions of the airspace may be less congested than others, a larger flexibility or room for maneuver can be given to airspace users, air navigation service providers and airports operating in sparse areas without degrading the overall performance, i.e., the total cost of delay, of the entire system. The notion of flight criticality herein proposed is not intended as a measure of the amount of downstream effect, but rather should be considered as a measure of the degree of flight flexibility, given by the amount of time granted to flights to perform their operations.

In the following, a new mathematical model that identifies a period of time, defined as "time window", during which each flight operation (i.e., take off, landing and entry into a sector) should be executed is presented. The problem of identifying time windows in ATFM is called ATFM problem with Time Windows (ATFMTW). The width of a time window reflects degree of flexibility granted to flights: the larger the time window, the greater the amount of slack time available to execute the corresponding flight operation. The smaller the time window, the more important it is that all support operations for each phase of the flight, e.g. maintenance, ground and flight crew activities, and ATC clearances, are coordinated and executed on time, i.e., the more critical the flight. Both the position of the time window (i.e., its starting time instant) and its width (i.e., the number of time instants that belong to it) univocally determine the time window, and they depend on the state of the air traffic system.

An example of definition of a time window is illustrated in Figure 3.3. Here, the departure of a flight is scheduled at time instant $t = 1$. This indicates that the model guarantees the ability of a flight to depart within a window of time that is fixed to the size of the time instant. If the time instants are narrow, e.g., 5' wide, the execution of the departure movement has to be operated very carefully due to the strict time constraints provided by the model. Most importantly, the model does not indicate what is going to happen if the flight does not depart within the assigned time instant. Using time windows of flexible size, on the other hand, it is possible to indicate the criticality of flights. In the considered case, it is known that the flight will still be able to depart at $t = 2$, i.e., with one time instant of delay. In case of 5' wide time instants, this corresponds to a 10' departure time window, which is easier to deal with than a 5' time window. After the time window closes, i.e., at $t = 2$, there is no guarantee on the availability of resources to operate the flight, i.e., the flight may incur in additional delay. The time window concept is consistent with the Single European Sky ATM Research (SESAR) program as it enhances the responsibility of airlines in the context of the Air Traffic Management (ATM) system [56]. In fact, airlines are driven to manage their operations in order to deliver their flights inside these temporal intervals, thus leading to improved planning and earlier detection of delays.

An optimal position of the time windows, which minimizes the total amount of delay, can be computed with any ATFM model. Time windows can be assigned using a deterministic optimization model. Therefore, the literature on deterministic models for ATFM is now analyzed. The ATFM problem was first formalized in 1987 by Odoni [57]. Since then, a plethora of mathematical models have been developed. Some of the first models to include ground and airborne delay, as well as airspace capacity, were proposed by Helme [58] and Lindsay *et al.* [59], but both were unmanageable due to size limitations. Bertsimas and Stock Patterson [47] proposed a readily solvable large-scale model which also includes – in addition to ground holding and airborne delay – speed

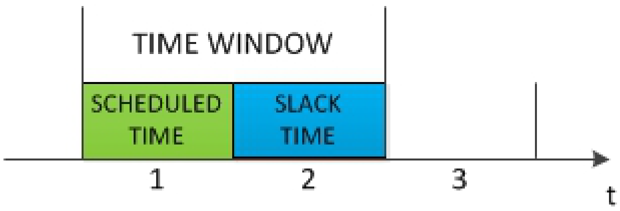

Figure 3.3: Example of time window

control as a control option. The authors showed that their model is NP-hard. The same authors later proposed a mathematical model [60] which also allows each flight to have more than one possible route, i.e., the possibility to reroute a flight. Alonso and Escudero [61] discussed the tightness of mathematical formulations for the ATFM problem. They compared the definition of variables that identify events either occurring *at* or *by* given time periods, and proved that the latter definition provides a tighter model. Several other authors addressed both delay and routing decisions simultaneously, including Sun and Bayen [62] in a control context and Myers and Kierstead [63] in a network flow context.

Lulli and Odoni [54] formulated an ATFM model that focuses on the characteristics of the European Union airspace. Their model has a cost function that ensures equity in the assignment of delays to flights. This is achieved by favoring the assignment of a moderate amount of delay to a large set of flights over the assignment of a large amount of delay to a limited number of flights. The intrinsic complexity of optimal ground and airborne holding strategies in the European context is also addressed. Their approach is consistent with the European specification of sector capacity, defined as the number of flights that can enter the sector in a single time period [64]. Recently, Bertsimas, Lulli, and Odoni [65] proposed a powerful and flexible Integer Programming model – in the following referred to as "BLO" – which combines the good mathematical properties of the Bertsimas-Stock Patterson model [47] with a complete representation of control actions. Their model optimizes the total cost of delay taking fairness issues into account. Three classes of valid inequalities are also included to improve the computational performance. Computational results are reported for two sets of instances, named *"regional"* (3,000 flights) and *"national"* (6,500 flights), that are solved within a 1% optimality gap in 305" and 743", respectively. An extension of this work was proposed by Churchill, Lovell, and Ball [66].

Agustín *et al.* [67] proposed a deterministic mixed binary model for ATFM that allows for flight cancellation and rerouting. Seven alternative objective functions that allow decision makers to achieve different goals are described. The proposed model is so tight that it does not require the execution of the branch-and-bound phase to obtain the optimal solution for any tested instance. Authors report computation times of 69" and 172" for a regional and national size instance of the BLO, respectively, as well as 394" to solve a modified (more difficult) BLO national size instance. Sun *et al.* [68] developed an aggregated ATFM model based on a multicommodity network to minimize the total travel time of flights in the entire U.S. airspace. The model is subject to sector, arrival and departure capacity constraints at an aggregated level, i.e., for the entire airspace rather than at each airport, for each time period. The authors relaxed their IP formulation into an LP for computational efficiency and used the dual decomposition procedure to solve the large-scale LP model. Around 95% of instances yield an IP solution from a relaxation solve and a rounding heuristic is used for the remaining instances. The computation time for instances that involve two-hour traffic flows for the entire U.S. airspace (around 8,000 flights) is around 150". In the following section, two alternative models for ATFMTW are proposed.

3.7 Deterministic optimization models

Time windows may be defined either jointly with the schedule or separately from it. The first approach, that is formulated in a model named "optimal model", is more computationally challenging. In fact, it does not simply look for a schedule with a minimum delay, but it determines which minimum delay cost schedule – as there may be multiple optimal schedules – provides the largest time windows. This model is illustrated in §3.7.1. The second approach, on the other hand, is formulated in a model named "near-optimal model". This is an easier approach, as it separates the decisions made on the schedule and on the width of the time windows. First, the first minimum delay cost schedule found is chosen as a solution which fixes the position of the time windows. Other possible minimum delay schedules are not considered. Then, the time windows are maximized for the chosen schedule, i.e., their width is determined. This model is presented in §3.7.2. Computational results for both models are discussed in §3.8. Finally, conclusions are drawn in §3.9.

3.7.1 Optimal time window model

In the optimal model, a decision is made on the flight schedule and the time windows at the same time, i.e., both their position and width is determined. This model is an extension of the BLO model, which provides a wide set of control options and is able to provide good computational results. In the following, the notation, decision variables and constraints that are used to formulate the optimal time window model are presented.

Notation

$$
\begin{aligned}
\mathcal{A} &\equiv \text{set of airports, indexed by } a \\
\mathcal{S} &\equiv \text{set of sectors, indexed by } s \\
\mathcal{F} &\equiv \text{set of flights, indexed by } f \\
S^f \subseteq (\mathcal{S} \cup \mathcal{A}) &\equiv \text{set of sectors/airports that can be flown by flight } f \\
\mathcal{T} &\equiv \text{set of time periods, indexed by } t \\
DC_a, AC_a, TC_a &\equiv \text{Set of departure, arrival and total capacity constraints for airport } a, \\
&\quad\ \text{indexed by } c\text{, regulating time periods in set } \tau_c \\
SC_s &\equiv \text{Set of capacity constraints for sector } s\text{, indexed by } c\text{, regulating} \\
&\quad\ \text{time periods in set } \tau_c \\
\mathcal{P} &\equiv \text{set of pairs of flights } (f, f') \text{ that are connected, with corresponding} \\
&\quad\ \text{turnaround time } l_{f,f'} \\
L_s^f &\equiv \text{set of sectors preceding sector } s \\
N_s^f &\equiv \text{set of sectors following sector } s \\
C_c &\equiv \text{available capacity for airport or sector constraint } c \\
orig_f &\equiv \text{airport of origin of flight } f \\
dest_f &\equiv \text{airport of destination of flight } f \\
t_{fss'} &\equiv \text{minimum number of time periods that flight } f \text{ must spend in sector } s \\
&\quad\ \text{before entering sector } s' \\
end_f &\equiv \text{maximum acceptable duration of flight } f \\
T_j^f = [\underline{T}_j^f, \overline{T}_j^f] &\equiv \text{set of feasible time periods for flight } f \text{ to operate at airport/sector } j \\
\underline{TW} &\equiv \text{minimum time window size} \\
\overline{TW} &\equiv \text{maximum time window size}
\end{aligned}
$$

Decision variables

Two sets of decision variables are defined, one for each type of decision to be made. The first set, named xs, is used to make decisions on the position, or start time, of time windows. The second set, named xe, is used to make decisions on the width of the time windows, by defining their closing time.

$$xs^f_{jt} = \begin{cases} 1, & \text{if the time window for flight } f \text{ at airport/sector } j \text{ has been opened} \\ & \text{by time } t \in T^f_j \\ 0, & \text{otherwise} \end{cases}$$

$$xe^f_{jt} = \begin{cases} 1, & \text{if time window for flight } f \text{ at airport/sector } j \text{ has been closed} \\ & \text{by time } t \in T^f_j \\ 0, & \text{otherwise} \end{cases}$$

Notice that, since flight cancellation is not considered in the model, it is possible to set variables $xe^f_{orig_f,\overline{T}_{orig_f}}$ and $xe^f_{dest_f,\overline{T}_{dest_f}}$ to 1 for each flight f. This ensures both the departure and arrival operations of flights within the feasible operation times specified. Furthermore, this definition of the decision variables, which indicates the opening or closing of a time window *by* instead of *at* a specific time instant, is consistent with the variable definition from the BLO model. This variable definition is consistent with the discussed findings by Alonso and Escudero [61].

Objective functions

Since the model makes two separate decisions, one on the position and the other on the width of time windows, two different objective functions are defined. The first objective, which is used to define the position of the time windows, is the minimization of the total cost of delay. For a flight f, this total delay cost is the sum of the departure and arrival delay costs. The cost coefficients $dc^f(t)$ and $ac^f(t)$ represent the cost of delay for flight f when the departure and arrival time windows open at time t. They are defined as follows:

$$\begin{aligned} dc^f(t) =& (t - \underline{T}^f_{orig_f})^{1+\epsilon_d} \\ ac^f(t) =& (t - \underline{T}^f_{dest_f})^{1+\epsilon_a} \end{aligned}$$

where the values $\epsilon_d > 0$ and $\epsilon_a > 0$ are two positive parameters which make the cost coefficients super-linear. As discussed by Lulli and Odoni [54], this choice grants a fair assignment of delay among different flights, favoring the distribution of delays among different flights rather than concentrating it on a small set of flights. The objective function minimizing the total cost of delay is formulated as follows:

$$Z_1 = Min \sum_{f \in \mathcal{F}} \left(\sum_{t \in T^f_{orig_f}} \Big((xe^f_{orig_f,t} - xe^f_{orig_f,t-1}) \cdot dc^f(t) \Big) + \right. \tag{3.3a}$$

$$\left. + \sum_{t \in T^f_{dest_f}} \Big((xe^f_{dest_f,t} - xe^f_{dest_f,t-1}) \cdot ac^f(t) \Big) \right)$$

Once the minimum delay cost Z_1 is determined, the second objective is to maximize the width of the time windows. This is done considering all minimum cost schedules. This way, it is possible to provide greater flexibility to the different stakeholders with no harm to the overall system performance. Mathematically, the second objective function looks for the optimal solution on the polyhedron made of the optimal solutions of the first step. The objective is to maximize the number of time periods composing each time window:

$$Z_2 = Max \sum_{f \in \mathcal{F}, j \in S^f, t \in T^f_j} (xs^f_{j,t} - xe^f_{j,t-1}) \tag{3.3b}$$

being subject to the additional constraint:

$$\sum_{f \in \mathcal{F}} \left(\sum_{t \in T^f_{orig_f}} \Big((xs^f_{orig_f,t} - xs^f_{orig_f,t-1}) \cdot dc^f(t) \Big) + \right. \tag{3.3c}$$

$$\left. + \sum_{t \in T^f_{dest_f}} \Big((xs^f_{dest_f,t} - xs^f_{dest_f,t-1}) \cdot ac^f(t) \Big) \right) = Z_1$$

Constraints

The optimal time window model has the following sets of constraints.

Capacity constraints

$$\sum_{f \in \mathcal{F}: a = orig_f, t \in \tau_c} (xs^f_{at} - xe^f_{a,t-1}) \leq C_c \quad \forall a \in \mathcal{A}, c \in DC_a \tag{3.3d}$$

$$\sum_{f \in \mathcal{F}: a = dest_f, t \in \tau_c} (xs^f_{at} - xe^f_{a,t-1}) \leq C_c \quad \forall a \in \mathcal{A}, c \in AC_a \tag{3.3e}$$

$$\sum_{f\in\mathcal{F}:a=orig_f\vee a=dest_f,t\in\tau_c}(xs^f_{at}-xe^f_{a,t-1})\leq C_c\quad\forall a\in\mathcal{A},c\in TC_a \tag{3.3f}$$

$$\sum_{f\in\mathcal{F}:s\in S^f,t\in\tau_c}(xs^f_{st}-xe^f_{s,t-1})\leq C_c\quad\forall s\in\mathcal{S},c\in SC_s \tag{3.3g}$$

Constraints (3.3d), (3.3e) and (3.3f) ensure that capacity is respected at airports for departure, arrival and total flight movements, respectively. Capacity constraints at air sectors are defined by (3.3g). Notice that this definition for sector capacity is consistent with the European definition of sector capacity.

Route constraints

$$xe^f_{j,\bar{T}^f_j}\leq\sum_{j'\in N^f_j}xe^f_{j',\bar{T}^f_{j'}}\quad\forall f\in\mathcal{F},j\in S^f:j\neq dest_f \tag{3.3h}$$

$$xe^f_{j,\bar{T}^f_j}\leq\sum_{j'\in L^f_j}xe^f_{j',\bar{T}^f_{j'}}\quad\forall f\in\mathcal{F},j\in S^f:j\neq orig_f \tag{3.3i}$$

$$\sum_{j'\in N^f_j}xe^f_{j',\bar{T}^f_{j'}}\leq 1\quad\forall f\in\mathcal{F},j\in S^f:j\neq dest_f \tag{3.3j}$$

$$\sum_{j'\in L^f_j}xe^f_{j',\bar{T}^f_{j'}}\leq 1\quad\forall f\in\mathcal{F},j\in S^f:j\neq orig_f \tag{3.3k}$$

$$xs^f_{jt}\leq\sum_{j'\in N^f_j}xs^f_{j',t-l_{fj'j}}\quad\forall f\in\mathcal{F},j\in S^f:j\neq orig_f,t\in T^f_j \tag{3.3l}$$

$$xe^f_{jt}\geq\sum_{j'\in L^f_j}xe^f_{j',t-l_{fj'j}}-(1\;-\;xe^f_{j,\bar{T}^f_j})\quad\forall f\in\mathcal{F},j\in S^f:j\neq dest_f,t\in T^f_j \tag{3.3m}$$

Constraints (3.3h) and (3.3i) represent the execution of a flight along a route, given by the sequence of contiguous sectors/airports crossed from origin to destination. Constraints (3.3h) link a sector/airport j to a following sector/airport $j'\in N^f_j$. Constraints (3.3i) ensure that a flight f can reach a sector/airport j if and only if a preceding sector/airport $j'\in L^f_j$ is included in the flight route. Constraints (3.3j) and (3.3k) ensure that a flight f only follows one route from origin to destination. Constraints (3.3j) guarantee that a flight that has reached some sector/airport j will only reach a specific successive sector/airport $j'\in N^f_j$. Constraints (3.3j), on the other hand, ensure that a flight f that has reached some sector/airport j can come from only one preceding sector/airport $j' \in P_j^f$. Constraints (3.3l) and (3.3m) stipulate that a flight f cannot enter a following sector j' along its route until it has spent at least $l_{fjj'}$ time periods (the minimum flight time) travelling through the current sector/airport j.

Connectivity and flight duration constraints

$$xs^{f'}_{orig_{f'},t+s_f} \leq xe^f_{dest_f,t} \quad \forall (f,f') \in \mathcal{P}, t \in T^f_{dest_f} \tag{3.3n}$$
$$xs^f_{orig_f,t} \leq xe^f_{dest_f,t+end_f-1} \quad \forall f \in \mathcal{F}, t \in T^f_{orig_f} \tag{3.3o}$$

The turnaround time between connected flights is guaranteed by constraints (3.3n), which separate the departure and arrival time windows of two connected flights. Constraints (3.3o) impose that the total flight time does not exceed the maximum duration allowed for the flight.

Time windows definition

$$xs^f_{jt} - xe^f_{jt} \geq 0 \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.3p}$$
$$xs^f_{j,\overline{T}^f_j} = xe^f_{j,\overline{T}^f_j} \quad \forall f \in \mathcal{F}, j \in S^f \tag{3.3q}$$
$$xe^f_{j,t+\underline{TW}-1} \leq xs^f_{jt} \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.3r}$$
$$xs^f_{j,t-\overline{TW}+1} \leq xe^f_{jt} \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.3s}$$

Constraints (3.3p) ensure that time windows are correctly formulated, i.e., if a time window opens, then it must be at least one time instant wide. Furthermore, constraints (3.3q) ensure that time windows that open must also close by the last feasible time period. Constraints (3.3r) and (3.3s) define the minimum and maximum size for a time window, respectively.

Decision variables definition

$$xs^f_{j,t+1} - xs^f_{j,t} \geq 0 \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.3t}$$
$$xe^f_{j,t+1} - xe^f_{j,t} \geq 0 \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.3u}$$
$$xs^f_{jt} \in \{0,1\} \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.3v}$$
$$xe^f_{jt} \in \{0,1\} \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.3w}$$

Constraints (3.3t) and (3.3u) define decision variables as monotone increasing variables. Finally, constraints (3.3v) and (3.3w) define the decision variables as binary.

3.7.2 Near-optimal time window model

The near-optimal model makes limited decisions compared with the optimal model. In fact, it is designed to only compute the width of time windows. The position of each time window is given by the optimal solution of an ATFM model of choice. Hence, the model determines the optimal closing time for all time windows such that their total width is maximized. This does not guarantee an optimal solution to the problem because only a specific flight schedule is considered, which may not provide the largest time windows at the minimum delay cost.

The complete formulation of the near-optimal model is presented in the following, illustrating its notation, decision variables, objective function and constraints. This notation is slightly different from the one used for the optimal model, to follow more closely the original definition of the BLO model that is used to generate optimal solutions to the ATFM problem. In particular, capacity constraints are defined at each time instant and only regulate capacity at that specific instant, rather than considering a set of contiguous time instants as in the optimal model.

Notation

$$
\begin{aligned}
\mathcal{A} &\equiv \text{set of airports, indexed by } a \\
\mathcal{S} &\equiv \text{set of sectors, indexed by } s \\
\mathcal{F} &\equiv \text{set of flights, indexed by } f \\
S^f \subseteq (\mathcal{S} \cup \mathcal{A}) &\equiv \text{set of sectors/airports that can be flown by flight } f \\
\mathcal{T} &\equiv \text{set of time periods, indexed by } t \\
\mathcal{P} &\equiv \text{set of pairs of flights } (f, f') \text{ that are connected, with corresponding} \\
&\ \text{turnaround time } l_{f,f'} \\
D_a^t, A_a^t, T_a^t &\equiv \text{available capacity for departures, arrivals and total movements at} \\
&\ \text{airport } a \text{ and time } t \\
S_s^t &\equiv \text{available capacity at sector } s \text{ and time } t
\end{aligned}
$$

$$\begin{aligned}
orig_f &\equiv \text{airport of origin of flight } f \\
dest_f &\equiv \text{airport of destination of flight } f \\
T_j^f = [\underline{T}_j^f, \overline{T}_j^f] &\equiv \text{set of feasible time periods for flight } f \text{ to operate at airport/sector } j, \\
& \text{ with } \underline{T}_j^f \text{ the opening time period of the time window} \\
\underline{t}_{fss'}, \overline{t}_{fss'} &\equiv \text{minimum and maximum number of time periods that flight } f \text{ must} \\
& \text{ spend in sector } s \text{ before entering sector } s'\text{, respectively}
\end{aligned}$$

Decision variables

Similarly to the optimal model, for each flight f, sector/airport j, and time period t the following set of monotone binary variables is defined:

$$x_{jt}^f = \begin{cases} 1, & \text{if time window for flight } f \text{ at airport/sector } j \text{ is still open at time } t \\ 0, & \text{otherwise} \end{cases}$$

In view of this definition, the decision variables are monotone decreasing, differently from the decision variables from the optimal model, which are defined as monotone increasing. Also, since the opening time of a time window is defined by input data coming from the execution of an ATFM model, the decision variables are only defined for sectors along the chosen route of a flight. Finally, if each time window needs to satisfy minimum width $\underline{TW}$ and maximum width $\overline{TW}$ constraints, it is possible to fix some variables to obtain this result. In fact, given the opening instant of a time window $\underline{T}_j^f$, the first constraint is satisfied by fixing $x_{jt}^f = 1$ for all time periods such that $t < \underline{T}_j^f + \underline{TW}$. Also, the second constraint is similarly satisfied by fixing $x_{jt}^f = 0$ for all time periods satisfying $t \geq \underline{T}_j^f + \overline{TW}$.

Objective function

The objective function is the maximization of total width of the time windows, which can be obtained as follows:

$$Max \sum_{f \in \mathcal{F}, j \in S^f, t \in T_j^f} \alpha_{jt}^f \cdot x_{jt}^f \tag{3.4a}$$

If all the α_{jt}^f coefficients are equal to 1, the objective function computes the total number of time periods assigned to time windows. However, considering sub-linear cost coefficients it is possible to include the notion of fairness, i.e., the model favors the assignment of time windows of similar width to multiple flights rather than assigning large time windows to a small number of flights, penalizing other flights. More specifically, given some $\epsilon > 0$, the α^f_{jt} coefficients can be defined as follows:

$$\alpha^f_{jt} = \begin{cases} \frac{(t-\underline{T}^f_j)^{1-\epsilon}}{t-\underline{T}^f_j}, & \text{if } t > \underline{T}^f_j \\ 0, & \text{otherwise} \end{cases}$$

Constraints

The formulation of the near-optimal model has the following sets of constraints.

Capacity constraints

$$\sum_{f\in\mathcal{F}:orig_f=a} x^f_{at} \le D^t_a \quad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{3.4b}$$

$$\sum_{f\in\mathcal{F}:dest_f=a} x^f_{at} \le A^t_a \quad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{3.4c}$$

$$\sum_{f\in\mathcal{F}:orig_f=a\vee dest_f=a} x^f_{at} \le T^t_a \quad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{3.4d}$$

$$\sum_{f\in\mathcal{F}:s\in S^f} x^f_{st} \le S^t_s \quad \forall s \in \mathcal{S}, t \in \mathcal{T} \tag{3.4e}$$

Departure, arrival and total capacity constraints are defined by (3.4b), (3.4c) and (3.4d), respectively. Similarly, constraints (3.4e) ensure that available capacity is respected at sectors. If a time window of a flight f is open at time period t, then one unit of capacity is reserved to the flight. This modeling approach guarantees that all flight phases can be executed with no additional delay if they occur within the assigned time window.

Time windows constraints

$$x^f_{jt} \le x^f_{j',t+\underline{t}_{fjj'}} \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.4f}$$

$$x^f_{jt} \ge x^f_{j',t+\bar{t}_{fjj'}} \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.4g}$$

Constraints (3.4f) and (3.4g) impose the consistency of the time windows. By taking into account the minimum and maximum aircraft speed, they prevent a given time window from being too wide if the time window at the subsequent or previous sector is narrow, and viceversa.

Connectivity constraints

$$x^{f'}_{orig_{f'},t} \geq x^{f}_{dest_f,t-l_{ff'}} \quad \forall (f,f') \in \mathcal{P}, t \in T^{f'}_{orig_{f'}} \tag{3.4h}$$

Constraints (3.4h) link the time windows of connected flights, providing the minimum spacing between the arrival of flight f at destination and the departure of flight f' from its airport of origin, with $dest_f = orig_{f'}$.

Decision variables definition constraints

$$x^f_{jt} \geq x^f_{j,t+1} \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.4i}$$

$$x^f_{jt} \in \{0,1\} \quad \forall f \in \mathcal{F}, j \in S^f, t \in T^f_j \tag{3.4j}$$

Constraints (3.4i) define the decision variables as monotone decreasing. Finally, constraints (3.4j) define the variables as binary.

This formulation can be extended to include other practical issues such as the maximum “total” amount of delay that an aircraft can incur over the course of its flight due to fuel limitations. Nevertheless, the flexibility granted to airlines allows them to take these and other operational aspects into account without explicitly adding further constraints to the model. In fact, it is under the airline’s responsibility to make sure that all its operations in support of the flight execution are fulfilled within the allowed level of slack time.

Capacity utilization

The mathematical model described above can lead to overly conservative solutions, reserving an excessive amount of capacity for each flight. Indeed, for any period of a time window, the model reserves all the capacity resources necessary for the execution of the corresponding flight operation. Suppose that a three-period time window is assigned to flight f. Then, with the considered definition of time windows, one unit of capacity is reserved to the flight for each time period, for a total of three units of capacity, even though the flight will actually use only one unit. In the following, this approach is referred to as the “conservative” criterion. To overcome this issue, it is possible to modify capacity constraints (3.4b)-(3.4e), introducing a "capacity utilization coefficient" β_t^τ for each period of the time window. This coefficient can assume values between 0 and 1, with 0 denoting that no capacity is reserved for the execution of the flight operation, and 1 denoting that capacity is granted. Given this definition, capacity constraints can be rewritten as follows.

Capacity constraints with utilization coefficient

$$\sum_{f\in\mathcal{F}:orig_f=a}\ \sum_{\tau\in T_a^f:\tau\geq t} \beta_t^\tau x_{a\tau}^f \leq D_a^t \quad \forall a\in\mathcal{A}, t\in\mathcal{T} \tag{3.5a}$$

$$\sum_{f\in\mathcal{F}:dest_f=a}\ \sum_{\tau\in T_a^f:\tau\geq t} \beta_t^\tau x_{a\tau}^f \leq A_a^t \quad \forall a\in\mathcal{A}, t\in\mathcal{T} \tag{3.5b}$$

$$\sum_{f\in\mathcal{F}:orig_f=a\vee dest_f=a}\ \sum_{\tau\in T_a^f:\tau\geq t} \beta_t^\tau x_{a\tau}^f \leq T_a^t \quad \forall a\in\mathcal{A}, t\in\mathcal{T} \tag{3.5c}$$

$$\sum_{f\in\mathcal{F}:s\in S^f}\ \sum_{\tau\in T_s^f:\tau\geq t} \beta_t^\tau x_{s\tau}^f \leq S_s^t \quad \forall s\in\mathcal{S}, t\in\mathcal{T} \tag{3.5d}$$

Different criteria can be defined by differently shaping the capacity utilization coefficient. For instance, it is possible to set the coefficient to the reciprocal of the width of the time window. This criterion is referred to as "proportional criterion". If a flight departure is associated with a five-period time window, for each period of the time window only $\frac{1}{5}$ of the capacity is reserved. This can be achieved defining the β_t^τ coefficients as follows:

$$\beta_t^\tau = \begin{cases} \frac{1}{t+1-\underline{T}_j^f}, & \text{if } \tau = t \\ \beta_\tau^\tau - \beta_{\tau-1}^{\tau-1}, & \text{if } \tau \geq t+1 \end{cases}$$

In Table 3.14, an example of definition of the β_t^τ capacity coefficients is shown for time windows of maximum width $\overline{TW} = 5$.

For the sake of clarity, consider the capacity constraint for sector s at time period $t = 2$, and a flight f such that $\underline{T}_s^f = 1$. The utilization coefficients for this constraint for flight f are given in the second row of Table 3.14. If the time window for the flight at this sector is still open at time period $\tau = 2$, the width of the time window is at least two time periods, and the capacity reserved for the flight is at most one half, thus explaining the coefficient $\beta_\tau^\tau = 1/2$. But, if the time window is also open at time period $\tau + 1$, its width is at least three, and the capacity utilization for the flight is at most one third. Since $1/3 = 1/2 - 1/6$, it follows that $\beta_\tau^{\tau+1} = -1/6$, and so on.

Table 3.14: PROPORTIONAL CRITERION: COEFFICIENTS FOR $\overline{TW} = 5$

	$\tau - \overline{T}_j^f$				
$t - \overline{T}_j^f$	0	1	2	3	4
0	1	-1/2	-1/6	-1/12	-1/20
1		1/2	-1/6	-1/12	-1/20
2			1/3	-1/12	-1/20
3				1/4	-1/20
4					1/5

Other capacity utilization criteria can be proposed. For instance, a third approach, called "intermediate criterion", guarantees the required capacity – one unit – for the first time period of the time window, while it employs a capacity utilization inversely proportional to the width of the time window for the remaining time periods. For this criterion, the capacity constraint coefficients are defined as follows:

$$\beta_t^\tau = \begin{cases} 1, & \text{if } t = \tau = \underline{T}_j^f \\ 0, & \text{if } t = \underline{T}_j^f \wedge \tau \geq \underline{T}_j^f + 1 \\ \frac{1}{t - \underline{T}_j^f}, & \text{if } t = \tau > \underline{T}_j^f \\ \beta_\tau^\tau - \beta_{\tau-1}^{\tau-1}, & \text{if } \tau \geq t+1 \end{cases}$$

Table 3.15 shows an example of definition of the capacity utilization coefficients using the intermediate criterion for $\overline{TW} = 5$.

Capacity constraints (3.5), which use the capacity utilization coefficients, can be considered as a surrogate relaxation of constraints (3.4b)-(3.4e). Therefore, it is possible that under the proportional and intermediate criteria, or under other user-defined criteria similarly defined, the actual demand at time t is larger than the available capacity. This phenomenon is analyzed in §3.8 using experimental data.

Table 3.15: INTERMEDIATE CRITERION: COEFFICIENTS FOR $\overline{TW} = 5$

	$\tau - \overline{T}_j^f$				
$t - \overline{T}_j^f$	0	1	2	3	4
0	1	0	0	0	0
1		1	-1/2	-1/6	-1/12
2			1/2	-1/6	-1/12
3				1/3	-1/12
4					1/4

3.8 ATFMTW Computational Results

In this section, the results from the computational experiments performed with both discussed models are presented. First, in §3.8.1, results are illustrated for tests performed using the optimal model. Then, in §3.8.2, results for the tests performed using the near-optimal model are presented.

3.8.1 Optimal model computational results

The computational experiments were performed on two sets of randomly generated instances on a computer with Intel Core 2 Duo CPU at 2.00 GHz and with 2GB of RAM. The mathematical model was implemented in Mosel modeling language, solving the instances with the Xpress solver. The two sets of instances are called "medium-size" and "small-size", and they only differ in the number of considered flights, which amounts to 750 and 350, respectively. In both sets, 50% of the flights are connected.

Instances are generated representing the airspace as a grid of squared cells. This choice allows to accommodate sectors of arbitrary shape. Airports are randomly located on the grid with a minimum distance between two airports of three cells. Airports are also randomly subdivided between regional and hub airports. A regional airport is only connected to hubs, while hub airports are connected to both types of airports. All instances involve 29 sectors and five airports, three of which are hubs. Flights are randomly generated, and their routes unfold along adjacent cells. The considered scheduling horizon spans over five hours and is subdivided into 15-minute time instants. The size of time windows can be equal to either one or two time instants. Capacity constraints at airports are only defined for departures and arrivals, with a limit of 35 flights every 30 minutes. At air sectors, the maximum number of entries within 30 minutes is also set to 35 flights. Other key parameters that are set for all flights are: the sector flight time, the maximum delay in the time window assignment, the departure and arrival delay cost coefficients, and the maximum extra-duration for the flight time.

Instances are studied with different sector and airport capacity levels, starting from nominal capacity and lowering it by 10% each time, to test the performance of the model under bad weather conditions – affecting the whole network – of different intensity. For each capacity level, ten different random instances are generated for medium-size instances, while for small-size instances 30 random instances were tested per capacity level. The optimal model performs two consecutive optimizations, therefore computation times are analyzed separately for each step. For the first step, i.e., delay minimization, feasible solutions are computed within a 1% optimality gap. For the second step, i.e., time windows maximization, feasible solutions are computed within a 5% optimality gap. Moreover, considering the constraints on the computation times at this decision phase, the time limit for the computation of both steps is set to 20'.

The computational results of the execution of the delay minimization step are reported in Tables 3.16 and 3.17 for the medium-size and small-size instance sets, respectively. Similarly, Tables 3.18 and 3.19 provide the computational results for the execution of the time windows maximization step for the two sets of instances. These four tables have the following structure. Column 1 (Capacity) shows the percentage of nominal capacity available for the considered instances. Column 2 (Sol. Time) provides the average computation time. Column 3 (Solved) provides the percentage of instances solved within the 20' time limit. Column 4 (Infeasible) shows the percentage of infeasible instances. Finally, column 5 (Opt. Sol.) provides the average value of the optimal delay cost.

Table 3.16: Delay minimization results for medium-size instances

Capacity	Sol. Time	Solved	Infeasible	Opt. Sol.
100%	312.0"	100%	0%	0
90%	448.1"	100%	0%	0
80%	744.3"	60%	0%	0
70%	1176.1"	20%	0%	33.6
40-60%	-	0%	0%	-
30%	-	0%	20%	-
0-20%	-	0%	100%	-

Table 3.17: Delay minimization results for small-size instances

Capacity	Sol. Time	Solved	Infeasible	Opt. Sol.
100%	48.6"	100%	0%	0
90%	49.4"	100%	0%	0
80%	59.8"	100%	0%	0
70%	68.7"	100%	0%	0
60%	66.0"	100%	0%	0
50%	108.3"	100%	0%	0
40%	276.3"	86.7%	0%	0
20-30%	-	0%	0%	-
0-10%	-	0%	100%	-

Table 3.18: Time windows maximization results for medium-size instances

Capacity	Sol. Time	Solved	Infeasible	Opt. Sol.	Gap
100%	231.0"	90%	0%	7254.2	3.6%
90%	231.2"	70%	0%	6946.0	3.4%
80%	242.9"	20%	0%	6613.0	3.4%
40-70%	-	0%	0%	-	-
30%	-	0%	20%	-	-
0-20%	-	0%	100%	-	-

Table 3.19: Time windows maximization results for small-size instances

Capacity	Sol. Time	Solved	Infeasible	Opt. Sol.	Gap
100%	27.5"	100%	0%	3597.6	2.6%
90%	31.8"	100%	0%	3601.5	3.2%
80%	33.4"	100%	0%	3602.8	3.3%
70%	57.0"	100%	0%	3567.1	3.1%
60%	44.8"	100%	0%	3486.2	3.4%
50%	124.9"	100%	0%	3578.3	3.9%
40%	293.8"	66.7%	0%	3514.0	4.1%
20-30%	-	0%	0%	-	-
0-10%	-	0%	100%	-	-

To summarize the results of the two optimization steps, an analysis is provided in Tables 3.20 and 3.21, which refer to the medium-size and small-size instances, respectively. The two tables have the following structure. Column 1 (Capacity) provides the percentage of the nominal capacity available. Columns 2 (Step 1 Time) and 3 (Step 2 Time) provide the average computation times for the first and second optimization steps, respectively. Finally, columns 4 (Step 1 %) and 5 (Step 2%) provide the percentage of total computation time spent on each optimization step.

Results for medium-size instances show that the computation times of delay minimization increase when available capacity decreases. The time windows maximization step,

Table 3.20: Global results for medium-size instances

Capacity	Step 1 Time	Step 2 Time	Step 1 %	Step 2 %
100%	312.0"	231.0"	57.5%	42.5%
90%	448.1"	231.2"	66.0%	34.0%
80%	744.3"	242.9"	75.4%	24.6%
70%	1176.1"	-	-	-

Table 3.21: Global results for small-size instances

Capacity	Step 1 Time	Step 2 Time	Step 1 %	Step 2 %
100%	48.6"	27.5"	63.9%	36.1%
90%	49.4"	31.8"	60.8%	39.2%
80%	59.8"	33.4"	64.1%	35.9%
70%	68.7"	57.0"	54.6%	45.4%
60%	66.0"	44.8"	59.6%	40.4%
50%	108.3"	124.9"	46.4%	53.6%
40%	276.3"	293.8"	48.5%	51.5%

on the other hand, does not follow this trend. However, a limited number of solutions is provided: no solution complete with time windows is determined within the 20' time limit for capacity reduced to 70% or less of the nominal. More solutions are available for small-size instances thanks to the reduced number of flights. For these instances, it is possible to observe the slow increase of solution times for both optimization steps when capacity decreases from 100% to 60% of the nominal. Computation times drastically increase for both steps when capacity decreases to 50% or less of the nominal.

Average reported computation times for both optimization steps are equal to 9'03" for medium-size instances when nominal capacity is available. These solution times are acceptable in practice, however the size of these instances is still small compared with real life problems. The implementation of larger instances was also tested, but the optimal model did not scale up well. In fact, instances with 750 flights, i.e., only 25 more flights than the considered medium-size instances, did not provide any integer solution for the first step after one hour of computation.

Another drawback of the proposed approach lays in the percentage of time spent computing the second step, i.e., maximum width time windows. As reported in Table 3.20, for instances with nominal capacity, over 42% of the computation time is spent maximizing time windows. This means that, after computing a minimum delay schedule, an additional time equal to 74% of the computation time already spent is needed in order to find an optimal schedule/time windows combination. This increase in computation times is not acceptable in practice, even reducing the computation times for the first step to those reported by most state-of-the-art models. These unsatisfactory results motivated the development of the near-optimal model, whose computational results are discussed in the next subsection.

3.8.2 Near-optimal model computational results

The computational experiments on the near-optimal model were conducted on a series of randomly generated instances whose dimensions are representative of problems of near-continental scope in Europe. More specifically, instances consider 30 airports (10 of which are hubs), 145 sectors, 50 time periods, and 6,475 flights. These figures are the same as the national size instances used to test the BLO model. Each instance includes bad weather fronts that reduce the capacity of a set of airports and sectors. Different levels of capacity reductions are studied. The time horizon is divided into 5' time periods. The width of each time window can vary from one to three time periods. The mathematical model is implemented in Mosel modeling language, using the Xpress IVE programming environment. The position of the time windows is determined with the BLO model. Computation times for instances of this size are reported by the authors to range between 3'18" and 19'40", depending on the capacity reduction level. The average computation times of the near-optimal model are very limited compared with those of the optimal model. In fact, the average computation time is 40", with very few instances requiring over 1' of computation for the most congested instances. These computation times are reported on a standard laptop, and constitute a small overhead after the computation of an optimal flight schedule. Therefore, the near-optimal model may feasibly be used in practice.

To analyze results, four sets of flights are identified in each instance. Each set involves flights with time windows of different size. The first set of flights, depicted in Figures 3.4-3.6 by a gray line with diamond indicators, considers flights with time windows that are one period wide from origin to destination. These are the "most critical" flights. The second set of flights, shown by the pink line with squared indicators, contains flights that only have time windows that are two periods (10 minutes) wide. The third set, shown by the red line with triangle indicators, only contains flights with time windows that are three periods wide, thus representing the flights with the largest degree of flexibility. Finally, a fourth group of flights with time windows of different widths, i.e., not constant along the route, is considered. These flights are referred to as "heterogeneous", and they are depicted by the black line with circle indicators in the figures.

Furthermore, the three capacity utilization criteria discussed in §3.7.2, i.e., the conservative, the intermediate and the proportional criteria, are considered. For each criterion, a diagram displaying the distribution of flights among the four sets under bad weather fronts of different intensity is reported in Figures 3.4 to 3.6. The intensity of the bad

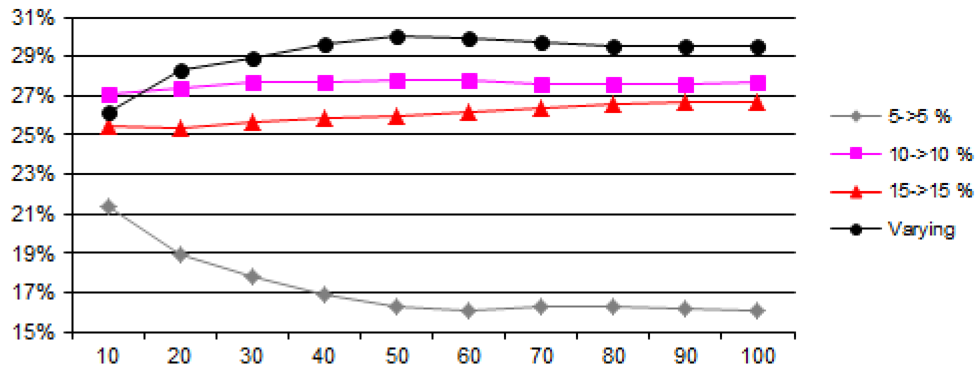

Figure 3.4: Flight distribution under conservative criterion

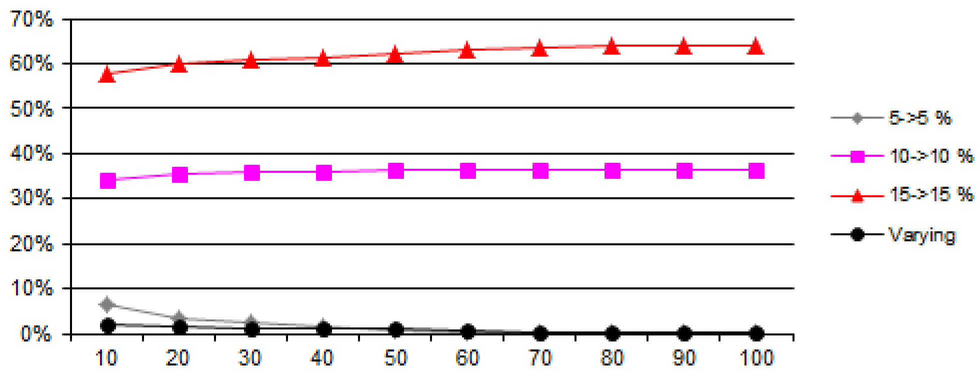

Figure 3.5: Flight distribution under proportional criterion

weather fronts is reported on the abscissa as the percentage of the available capacity with respect to the nominal capacity. For example, the value 10 represents the case with only 10% of the nominal capacity available at sectors and airports involved by the bad weather fronts, while the value 100 corresponds to the absence of bad weather fronts.

As expected, Figures 3.4-3.6 show that the number of most critical flights decreases when the available capacity increases. The percentage of most critical and heterogeneous flights shows a wide variability across the capacity utilization criteria. In fact, for the instances with the largest amount of available capacity, the percentage of most critical flights varies from 0% (proportional criterion, see Figure 3.5), up to 17% (conservative criterion, see Figure 3.4), whereas heterogeneous flights are always around 30%. These statistics support the observation that the initial formulation is indeed conservative in nature. Inversely, the implementation of constraints (3.5a)-(3.5d) implies that the

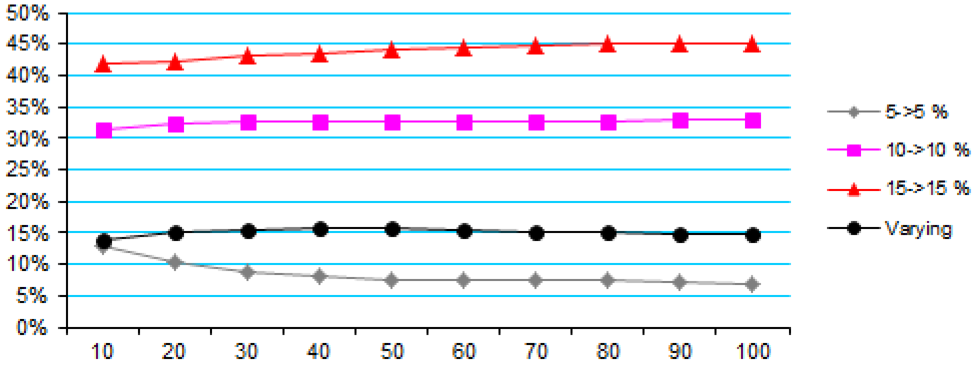

Figure 3.6: FLIGHT DISTRIBUTION UNDER INTERMEDIATE CRITERION

proportional capacity utilization criterion is quite inattentive in guaranteeing the required capacity resources for all flight operations. Under this criterion, the percentage of both the most critical and heterogeneous flights is almost irrelevant, especially for instances with a low level of congestion. A compromise solution is given by the intermediate criterion, where the sum of the percentages of the most critical and heterogeneous flights lies between 20% and 30%. The application of the proportional and intermediate criteria might lead to capacity shortages which should be addressed at the operational phase by assigning ATC delays. This calls for an evaluation of the robustness of the solutions, which is discussed in §3.8.3.

Figure 3.7 depicts the distribution of the time window width for heterogeneous flights under the conservative and intermediate criteria at three different levels of capacity availability: 10%, 50%, and 100%. The number of heterogeneous flights is negligible under the proportional criterion. As expected, the percentage of three time periods wide time windows increases with the available capacity for both criteria. Viceversa, the percentages of two and one time periods wide time windows decrease. Finally, the percentage of three time periods wide time windows is always slightly lower under the intermediate criterion. Nevertheless, their overall number is significantly higher than under the conservative criterion, as the percentages of flights with three time periods time windows only under the intermediate and conservative criteria are around 40-45% and 25%, respectively.

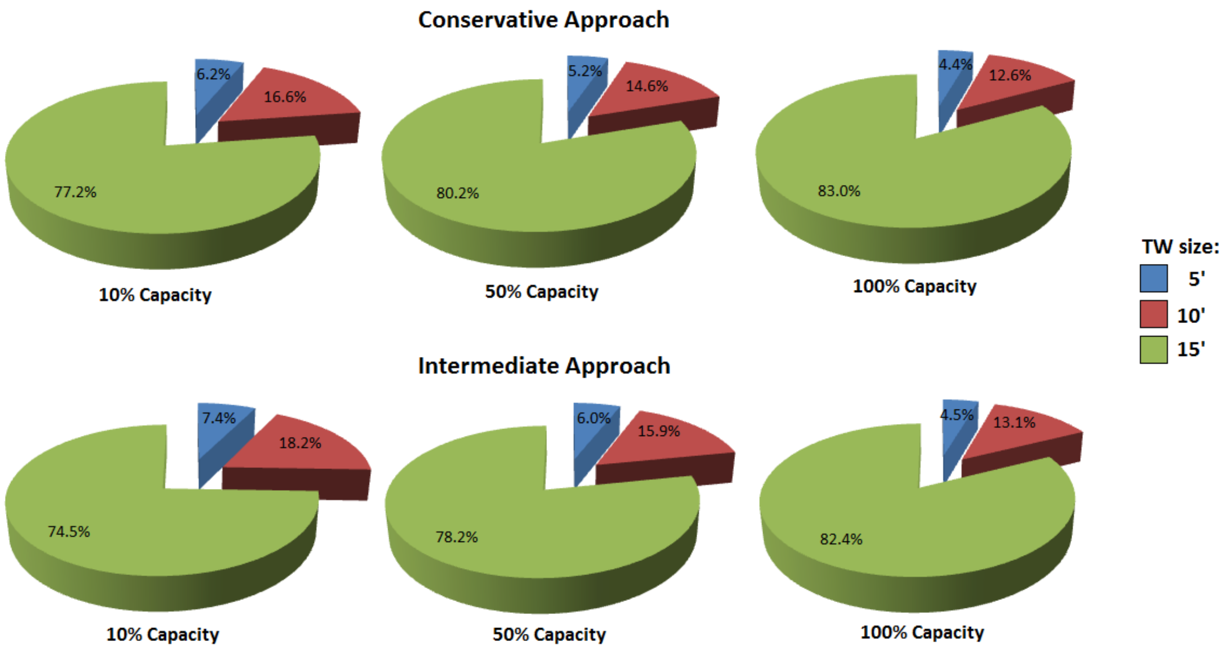

Figure 3.7: Time windows distribution for heterogeneous flights

3.8.3 Ex-Post Feasibility Analysis

Once the time windows for all flights are identified, each flight is entitled to use any of the available time periods within a time window to execute the corresponding operation. Therefore, discounting the use of capacity, as in the intermediate and the proportional criteria, may result in the violation of some of the capacity constraints due to an excess of the demand. If this adverse situation takes place, control actions have to be put into place with negative effects on the performances of the air traffic system. The likelihood of occurrence of these adverse situations under both the proportional and the intermediate criteria is evaluated. Capacity utilization is evaluated at every time period by simulating resource occupancy as follows. The departure time of each flight is randomly generated within its departure time window. Then, the sequence of time periods used by the flight along its route is determined. This procedure allows to compute the capacity demand for each time period and airspace element (airport/sector), and consequently to evaluate whether capacity constraints are violated or not. The random choice of the departure times is made in accordance with three probability distribution functions.

- **Uniform distribution:** all the time periods of a time window can be chosen with equal probability.
- **Triangular distribution:** the probability monotonically decreases with time. Hence, the first time period has the highest probability and the last time period the lowest one.
- **Mixed distribution:** the initial time period has probability $\frac{1}{2}$ to be chosen, whereas all the other time periods equally share the remaining probability.

For each congestion level, probability distribution, and capacity utilization criterion, 1,200 simulations of the usage of time windows were performed. Test instances involve about 50 time periods and 200 airports/sectors, therefore approximately 10,000 (airport/sector, time period) pairs are considered, each defining a capacity constraint. Table 3.22 shows the main results of this ex-post feasibility analysis, and it is structured as follows. Column 1 (Capacity) shows the level of available capacity at airports/sectors involved by a bad weather front. The figures in the other columns show how many (airport/sector, time period) pairs – out of 10,000 – report capacity violations under the corresponding probability distribution function for setting the random departure times. These figures are the average values over the 1,200 random tests performed.

The results show that, for each capacity level and probability distribution function, the number of capacity violations is on average always higher under the proportional criterion. These findings are coherent with the discount capacity factors used to define the two criteria. Nevertheless, the absolute values of the capacity violations are always extremely low under the proportional criterion too. In fact, the highest number of capacity violations – which occurs in the case of minimum capacity availability with departures chosen with the uniform distribution – is equal to 14.1 on 10,000 constraints, i.e., only 0.141% of capacity constraints is violated in the worst analyzed case.

The results of this analysis demonstrate the applicability of the proposed approaches, as the adopted criteria rarely produce adverse capacity shortages. In fact, even under the proportional criterion that guarantees the maximum level of flexibility to flights, with an almost irrelevant number of most critical flights (see Figure 3.5), capacity violations are very rare. Then, it is possible to conclude that the flexibility granted to flights has a negligible impact on capacity utilization even in congested situations, and thus does not produce any significant additional ATC delay.

Table 3.22: Feasibility analysis

	Proportional			Intermediate		
Capacity	Uniform	Triangular	Mixed	Uniform	Triangular	Mixed
10%	14.1	10.3	11.9	4.1	1.7	3.4
20%	10.5	8.7	9.7	2.5	0.9	2.4
30%	7.1	6.1	6.8	2.2	0.7	2.1
40%	5.7	5.2	5.6	2.0	0.5	1.9
50%	5.4	5.2	5.4	1.7	0.3	1.6
60%	4.3	4.1	4.3	1.8	0.4	1.7
70%	3.6	3.4	3.7	1.8	0.4	1.7
80%	2.7	2.8	3.1	1.7	0.3	1.6
90%	2.4	2.4	2.6	1.5	0.3	1.5
100%	2.3	2.3	2.5	1.5	0.3	1.4

3.9 ATFMTW Conclusions

In ATFM, time windows address the implicit uncertainty on the time of departure of flights. They can be used to measure the degree of flexibility of flight operations, allowing to identify flights whose execution is critical. Two different deterministic models are proposed in this chapter to determine time windows in ATFM. The first, called "optimal model", provides the optimal schedule/time windows combination. The second, called "near-optimal model", provides the largest time windows for a specific schedule, and is faster to compute.

Computational experiments on the optimal model were performed on small-scale random instances with moderate or low congestion. Computing time windows with this model is not trivial. In fact, for the largest instances considered, computing time windows after determining optimal schedules increases computation times by 74% in the least congested case. This increase in computation times is not acceptable for practical applications. This motivated the development of the near-optimal model, which allows to reduce computation times, making the time windows tool usable in practice. This model fixes the position of time windows, i.e., the initial instant, using any deterministic ATFM model from literature. Then, the total width of time windows is maximized for the chosen schedule. Instances of near-continental scope in Europe reported computation times to maximize time windows of 40" on average, showing the practical viability of the proposed approach.

The standard time window approach reserves capacity for all time instants within a time window. However, this approach can be too conservative in practice, resulting in the declaration of too many flights as critical. For this reason, two additional criteria – called "proportional" and "intermediate" criteria – for computing capacity utilization are proposed. Each criterion gives rise to a tradeoff between the level of conservatism and the occurrence of imbalances between demand and capacity when flights are operated. Computational results show that the proportional criterion identifies a very limited number of critical flights, while the intermediate criterion provides a more balanced solution, as each flight is always granted the possibility to operate at the initial time period of its time windows. Demand/capacity imbalances are evaluated by randomly choosing the departure time of a flight within the available time periods of its departure time window. Results show that, on average, capacity constraints are respected with a rate always higher than 99.8%, even when severe bad weather fronts are present, thus proving the viability of the proposed framework.

Interesting future developments of this work can be achieved by applying time windows to real air traffic data. Using real data, it may be possible to provide interesting insight on congestion phenomena through the identification of critical flights. In particular, the study of the spatial distribution of the width of time windows can allow to identify spacial bottlenecks in the system. Finally, uncertainty on both capacity and flight departure times may be considered together, joining the two different approaches discussed in this chapter, proposing time windows for stochastic solutions.

Chapter 4
OPERATIONAL DECISION MAKING

After the strategic and tactical decision phases, the final phase considered is the operational decision phase. In this chapter, the problem of rescheduling flights in real time is faced. Flight operations were scheduled at the tactical phase, and time windows to manage them were also determined. The application of time windows at the operational phase is now analyzed. The studied operational problem is named Real Time Flight Rescheduling with Time Windows (RTFRTW), and it is described in §4.1. To solve it in real time, a rule-based algorithm, described in §4.2, is used. The algorithm was tested on two sets of instances, one with simulated data and the other based on real flight data. These results are presented in §4.3. Finally, in §4.4, the conclusions and future directions of work on RTFRTW are summarized.

4.1 Real Time Flight Rescheduling with Time Windows

At the tactical phase, complete flight plans for the execution of flights are determined. As highlighted in §3.6, the operation times specified in the flight plans can be associated with time windows, which specify the slack time available to execute each flight operation without the need to ask for the assignment of new capacity resources. This is particularly critical at departure, as unforeseen events at the airport of origin of a flight involving passengers, staff or machines may take place. Time windows allow to identify flights that need to be executed carefully to avoid incurring in additional delays due to the lack of resources needed to operate them at a delayed time. Such flights are defined as “critical flights”.

Time windows can be used as a tool to deal with flight flexibility. However, they only measure the flexibility of each flight operation, without specifying the actions that should be taken when a flight is not able to respect a time window. In fact, time windows only specify that, in such a case, capacity resources are not assigned to the flight. No information is provided on when resources will be available for the flight, i.e., when it is possible to reschedule it. Such decisions can be made at the operational phase. At this phase, the uncertainty on the departure time of a flight is vanishing, and it is possible to determine the actual usage of capacity resources by flights. In fact, uncertainty on the departure time affects all flights, and the delayed departure of a flight can result in the ability of another flight to use the capacity resources initially scheduled to be used by the first flight. A framework to reschedule flight operations at the operational phase is the subject of this chapter.

The proposed framework is based on an algorithm that reschedules flights in real time according to the evolving capacity availability in order to reduce delays. This algorithm follows the principles of the Ration-By-Schedule (RBS) procedure [69]. RBS rations limited capacity resources according to schedule-based priority rules. Rationing is typically done under the Collaborative Decision Making (CDM) framework [70, 71]. CDM in Europe is concerned with the data exchange between the Central Flow Management Unit (CFMU) and the airlines. Data exchange is a key factor leading to better decision making in ATFM. CDM also aims at the decentralization of decisions which have a potential economic impact on airline operations, to make them in collaboration with the airlines. Under CDM, by rationing available capacity, slots are assigned to airlines rather than to flights. Then, airlines may perform substitutions or cancellations of slots, resulting in a slot usage different from the rationed one. It is therefore necessary to execute a compression algorithm to avoid wasting unused capacity resources. This is achieved by moving flights earlier in time to fill slots that are available after the substitution/cancellation.

The RBS algorithm was introduced to substitute the Grover Jack algorithm, see Ball *et al.* [72]. The Grover Jack algorithm forms a list of flights ordered by the most recent estimated time of arrival of each flight. Controlled arrival times are assigned following this list, depending on the arrival slots available at the destination airport. However, this mechanism is subject to several problems. First, a so-called “double penalty” can occur. In fact, an incoming flight reporting its delay is penalized in the Grover Jack flight list, and may receive additional delay, which would have not been assigned if the airline did not report the original delay. Also, slots allocated to cancelled flights are allocated to other flights regardless of the operating airline, which resulted in airlines not reporting flight cancellations to avoid favoring rival airlines. The RBS mechanism was introduced to overcome these problems. In literature, different issues related to RBS and resource allocation under CDM are faced. Vossen [73] discussed fair allocation of resources in CDM procedures, introducing and evaluating several approaches to fair allocation, using both multi-objective optimization models and cooperative models from game theory. Ball and Lulli [74] investigated how the set of flights to which delays are applied under CDM is defined. They defined a "distance-based" GDP that only applies to flights coming from airports within a prescribed distance from destination, and they studied how to appropriately define this distance. Brinton *et al.* [75] discussed an airport surface departure management concept that was evaluated operationally in the field in the U.S., the Collaborative Departure Queue Management (CDQM) concept. An application from Memphis, TN was studied and CDQM is reported to provide reduced taxi times, reducing fuel usage while maintaining full use of departure capacity. Jones *et al.* [76] studied four different algorithms designed to reduce the amount of delay in the terminal phase of flight. Different priority criteria are evaluated, with three algorithms being variants of the RBS algorithm, while the last uses an integer program to assign arrival times. Their analysis suggests that RBS-based algorithms provide strong throughput performance, which is achieved at the expense of higher fuel usage.

Another issue involving slot allocation under CDM is given by slot exchange between airlines. Vossen and Ball [77] discussed how the procedures for allocating slots to airlines and exchanging slots between airlines may be formalized through appropriately defined optimization models. In addition, they described how inter-airline slot exchanges may be viewed as a bartering process, in which each "round" of bartering requires the solution of an optimization problem. They compared the resulting optimization problem with the existing procedure for exchanging slots and discussed possibilities for increased decision-making capabilities by the airlines. Auction mechanisms are also studied. Cholakerin [78] proposed a sequential, sealed-bid Vickery auction without package bidding as a mechanism for allocating arrival slots during Ground Delay Programs (GDPs). The auction was simulated on historical flight data and compared with two CDM slot auction methods and a "global optimization" method where it is assumed that one airline owns all flights. Computational results report that the proposed auction mechanism improves the performance of CDM. Balakrishnan [79] considered two different approaches based on market design to resolve the issue of airport landing resource allocation. His work analyzes the problem of slot trading without monetary payments, presenting sufficient conditions for the existence of stable allocations. Furthermore, a combination of optimization-based slot trading with a payment-based exchange scheme that uses the Vickrey-Clarke-Groves (VCG) pricing mechanism [80] is proposed.

Many literature works solve problems that, despite being similar to the operational problem faced in this chapter, are more tactical in nature. This allows to use optimization models, algorithms or auction methods that are not compatible with real time requirements. Operational actions are usually faced by Air Traffic Control (ATC), which is a service provided by ground-based controllers that manage airborne flights enforcing flight safety rules. Auction mechanisms compatible with operational problems may be proposed. A possible mechanism is a protocol-based auction mechanism with time limit managed by the CFMU. In this mechanism, airlines may submit bids at any time before the deadline, being aware of the current winning bid, thanks to a continuous communication between the airlines and the CFMU. However, this mechanism may be subject to issues related to slow bid increase. An alternative mechanism may be inspired by token ring networks (see the IEEE 802.5 standard). In such a mechanism, airlines place bids at turns defined by the ring, with each airline having limited time to place a bid. However, this mechanism may be slow when many airlines participate in the auction, resulting inefficient in practice. A solution to this inefficiency may be provided by a first-price sealed-bid auction mechanism [81]. In this mechanism, airlines do not need to send a message to enter the auction, as they are free to send their own offer by some deadline. When the deadline expires, the CFMU announces the winner of the auction. Very short times can be granted for the auction. However, it is known that such auctions give the bidders incentive to bid lower than their valuation of the item, lowering the final revenue.

This chapter focuses on the application of RBS and compression algorithms to the special case of time windows. Auction mechanisms are not considered. The difference with RBS mechanisms from literature is in the reservation of slots. In fact, using time windows, flights may be assigned multiple consecutive slots, reflecting the size of their time windows. Also, airlines behavior is influenced by the existence of time windows, as when they are respected flights do not need to be rescheduled through capacity compression. In the proposed framework, slots are assigned to flights instead of airlines, providing a maximum system flexibility. This may however be difficult to implement in practice due to the unwillingness of airlines to release resources that may favor their competitors. Collaborative and non-collaborative airline attitudes are explicitly faced in this work, to study the global benefits resulting from collaborative airline attitudes.

The starting point of this problem is the a schedule with time windows, defined as in chapter 3. This schedule may contain ATFM delays applied to flights. When the scheduled departure time – that includes the possible ATFM delay – of a flight approaches, its operating airline is, in general, able to reduce the uncertainty on the actual time of departure of the flight. Some time before the scheduled departure time, the airline is supposed to be able to determine an accurate final departure time for the flight. A delay on the scheduled departure time may have to be applied. This delay is referred to as "additional delay". The additional delay may cause the flight to exceed its departure time window. Supposing that time windows are defined according to the conservative criterion (see §3.8), an airline that expects the departure of a flight to take place within the assigned departure time window does not need to take any corrective action. On the other hand, an airline that expects a flight not to meet the assigned departure time window needs to request for the assignment of new capacity resources to depart at a delayed time. If capacity is not available at the new requested time, a new delay should be applied. This delay is referred to as "increased additional delay".

Defining time windows conservatively reduces the amount of available capacity resources to operate flights. By definition, capacity resources are assigned to a flight at each time instant of a time window that is defined according to the conservative criterion. This can result in a situation such as the one illustrated in Figure 4.1. In this example, four flights are assigned three time periods wide time windows to depart. At time period $t = 3$, departure capacity is congested even if only a single flight is scheduled to depart at that time, due to the time windows of all three other flights being open at that time period. From a tactical point of view, this means that all flights may freely depart at time period $t = 3$. However, from an operational point of view, this may result in capacity shortages. For example, another previously scheduled flight may need to delay its departure at $t = 3$. Due to the presence of time windows, it may not be possible to reschedule its departure at that time period.

RTFRTW deals with the discussed problem, proposing a way to keep time windows without incurring in relevant capacity shortages for flights not respecting them. During flight execution, capacity availability is updated at every time instant. This is done by releasing capacity resources that are not reserved by flights anymore. This happens at the departure of a flight, when it is possible to close its departure time window as the

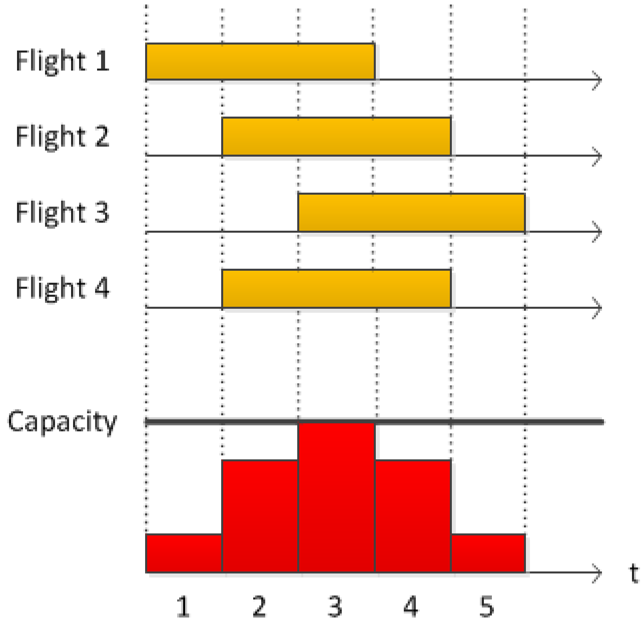

Figure 4.1: Capacity availability example

flight does not need resources to depart anymore. Capacity may also be released when an airline requests for a new departure time for a flight outside its assigned time window, as it is not able to use the allocated resources as scheduled. New capacity availability is then compared with requests for capacity usage by airlines for delayed flights. Capacity is allocated according to precedence rules which favor flights rescheduled with a large increased additional delay. Flights are rescheduled to single time periods, i.e., no slack time is provided to operate at the new requested departure time.

The described approach does not involve any collaboration from airlines, as they only provide information when they have a direct interest in doing so. To improve the performance of the global system, a different approach is proposed and evaluated. This new approach is named “collaborative approach”, while the approach described so far is referred to as “non-collaborative approach”. The collaborative approach requires airlines to provide the CFMU with the planned time of departure of a flight before this operation takes place, even when a flight is scheduled to depart within its departure time window. This allows to restrict time windows before departures take place, so capacity resources are available some time periods in advance compared with the non-collaborative approach. It is therefore easier to reschedule a flight at a new desired time of departure.

The described procedure is applied in real time using a rule-based algorithm based on RBS. The strict constraints on computation times do not allow to use optimization models at this decision phase: as soon as some resources are released by an airline, they should be reallocated in real time, if a reallocation is possible. The proposed algorithm is described in the following section.

4.2 Real time capacity reassignment algorithm

The RBS-based algorithm to reassign capacity resources to flights in real time is illustrated in Figure 4.2. Input data for the algorithm are the flight plans with the associated time windows and the capacity constraints for airports and sectors. This information is given for the time horizon under analysis. After reading the input data, a loop starts, with one iteration per time period from the considered time horizon. At each time period, airlines submit requests for new departure times to allocate to flights that are expected not to meet the assigned departure time windows. Also, flights that have departed are not considered for reallocation anymore. In the collaborative approach case, announced final departure times are received for all flights.

In the block denoted with (a), flights for which a reallocation request is formulated are assigned an earliest arrival time given the available capacity resources. If a flight is not assigned the new requested departure time, it is added to a pool of flights named *toReallocate*. For these flights, the algorithm will later try to reallocate their departure time at a time that is closer to their request. This is one of the crucial parts of the algorithm, and it is described into detail in §4.2.1. Capacity resources are then released for all operations which do not need them anymore. First, flights that have departed are assigned time windows of unitary size along all the route, so the corresponding time windows can be restricted. This holds only in the non-collaborative case. In the collaborative case, at this phase, time windows are already of unitary size. In fact, time windows for flights that announce a final departure time within the departure time window are restricted to unitary size at the time of announcement of the final departure time. Under the collaborative approach, airlines are requested to announce the final departure time of flights by a deadline. This deadline is fixed to a predefined amount of time before the scheduled departure time of each flight. Furthermore, resources previously assigned to flights requesting for a new departure time are also released.

After the release of all these resources, the block denoted with (b) in the flowchart is executed. This is the most important step of the algorithm, as it performs the reallocation

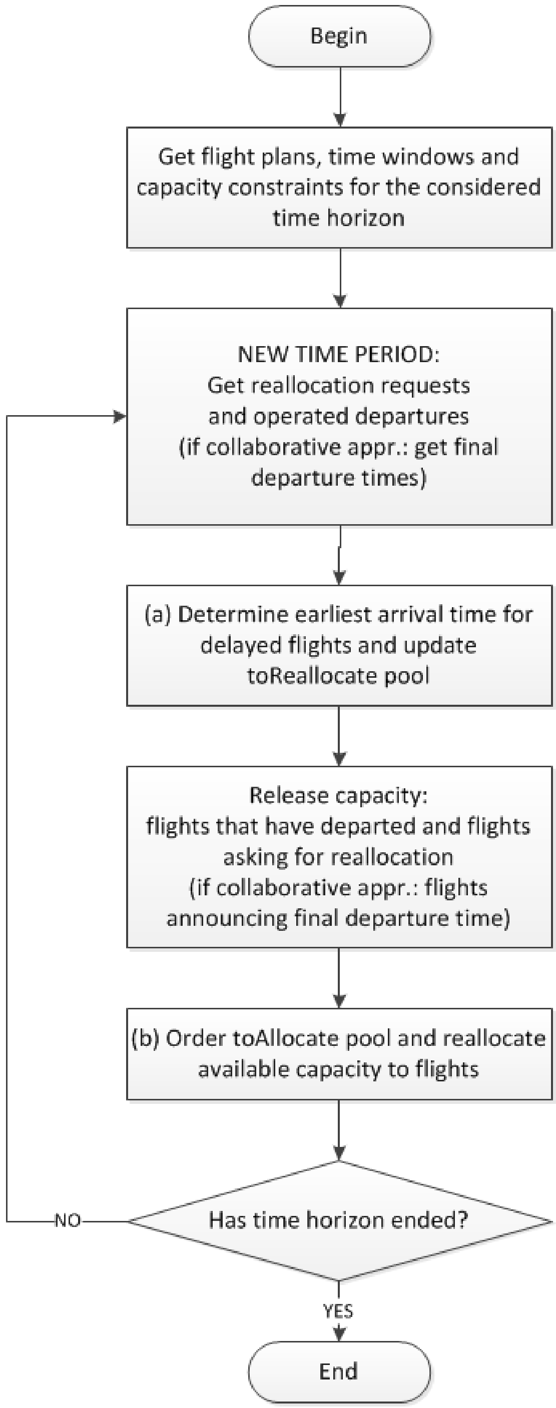

Figure 4.2: ALGORITHM FOR RTFRTW

of capacity resources. The details of this block are illustrated in §4.2.2. First, the *toReallocate* pool of flights is ordered according to predefined rules. Then, available capacity resources are reallocated to flights that belong to this pool, according to the defined order. The described steps are repeated for each time period of the considered time horizon, until the final one, then the algorithm stops.

In the next two subsections, blocks (a) and (b) are discussed into detail. Then, in §4.2.3, an example of the execution of the algorithm is provided.

4.2.1 Management of new reallocation requests

In Figure 4.2, block (a) refers to the management of new requests to operate flight departures at a time period outside the assigned time window. Details on the functioning of this block are shown in Figure 4.3. All reallocation requests are placed in a set named *reallocateRequests*. Each element contains information on the new requested time of departure for a flight. All elements of the set are analyzed in a loop. After the selection of a flight from the set, a variable t is set to the requested departure time, and the availability of the complete set of capacity resources needed along the route to depart at t is checked. If capacity is insufficient to operate the flight at t, the algorithm looks for the earliest departure time available by incrementing t until available resources are found. The flight plan can then be updated, reserving the needed resources. If the allocated departure time matches the request, the flight is removed from the *reallocateRequests* set. Else, the flight is moved to the *toReallocate* set, whose flights are considered for rescheduling later, to decrease their increased additional delay. These operations are repeated for each element of the *reallocateRequests* set, until it is empty.

4.2.2 Resource reallocation

Block (b) from Figure 4.2 is the core of the real time flight rescheduling algorithm, as it reallocates available capacity to flights. The detailed flowchart is provided in Figure 4.4. Only flights that are scheduled to depart at least two time periods from the current instant are considered for resource allocation. In fact, reallocating a flight scheduled to depart in the next time period implies rescheduling it at the current time period, forcing it to depart immediately, with too little time to react to the updated decision. Flights not satisfying this requirement are therefore removed from the *toAllocate* set, which is the set of all flights that are considered for rescheduling, to decrease their increased additional delay.

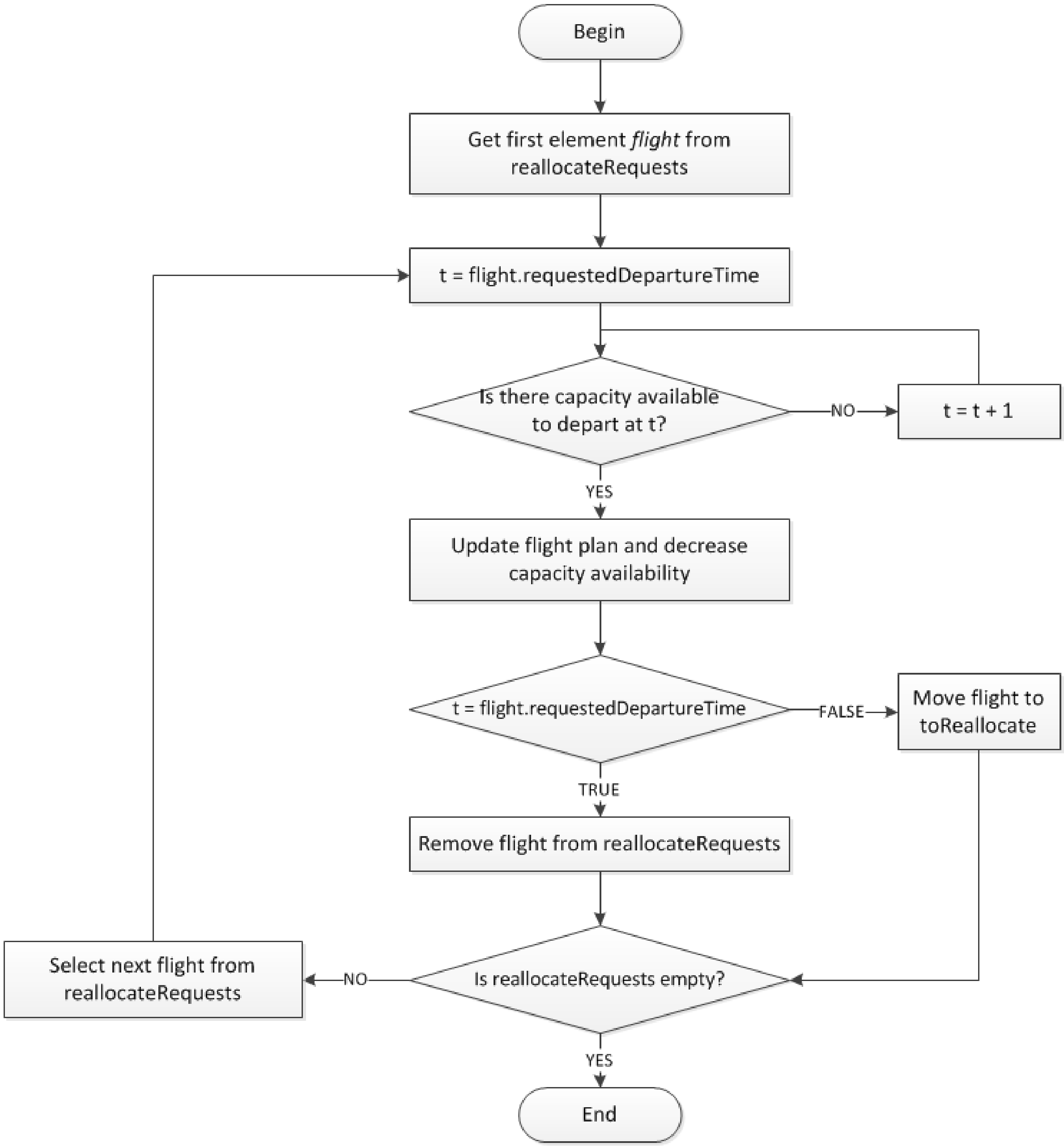

Figure 4.3: MANAGEMENT OF NEW REALLOCATION REQUESTS

The second step of the algorithm, which starts the main cycle, orders the *toAllocate* set according to the specified precedence rules. Ordering is performed according to the two following rules:

1. flights with a larger arrival delay get the precedence;

2. flights with an earlier scheduled arrival time get the precedence.

After ordering the flights, a variable named *reallocated* is set to 0. This variable counts how many flights have been reallocated in the current iteration. A reallocation involves the release of resources that were assigned to a flight that is now reducing its delay. These resources may now be used by some flight that has already been analyzed in the cycle. Using the *reallocated* variable it is possible to check whether some new reallocation may be possible in a new iteration, which is performed if $reallocated > 0$.

After initializing the *reallocated* variable, the first flight from the *toAllocate* set is retrieved, to start analyzing all flights in an inner cycle. The inner cycle starts by setting the first time period at which reallocation of the selected flight's departure is attempted. The value t is therefore set to the maximum value between the requested flight time and the time period following the one at which the algorithm is executed. No earlier time period is feasibly assignable to the flight to depart. Then, capacity availability at t is verified. If capacity resources are insufficient to move the departure at t, the same test is performed at the following time instant, until a feasible better reallocation is found, if one can be found. In fact, reallocations are looked up to the time instant at which t is less than the previously assigned departure time. If no better reallocation is found, the algorithm moves on to the following iteration of the inner cycle, selecting the next flight from the *toReallocate* set. On the other hand, if a better reallocation is found, the assigned departure time and the complete flight plan are updated. Then, capacity resources related to the old flight plan are released, and those related to the new flight plan are reserved. Also, the *reallocated* variable is incremented, to count the performed reallocation. If the flight is reallocated to the earliest possible departure time, it can be removed from the *toReallocate* set, since no better reallocation can be performed. Then, the algorithm goes to the next iteration of the inner cycle, by selecting the next flight from the *toReallocate* set. Once all flights from the *toReallocate* set are analyzed, the inner cycle ends. The main cycle now also ends, checking the value of the *reallocated* variable: if $reallocated > 0$ and *toReallocate* is not empty, then some additional improvement may be achieved. Therefore, the main cycle iterates. Else, the reallocation part of the algorithm terminates.

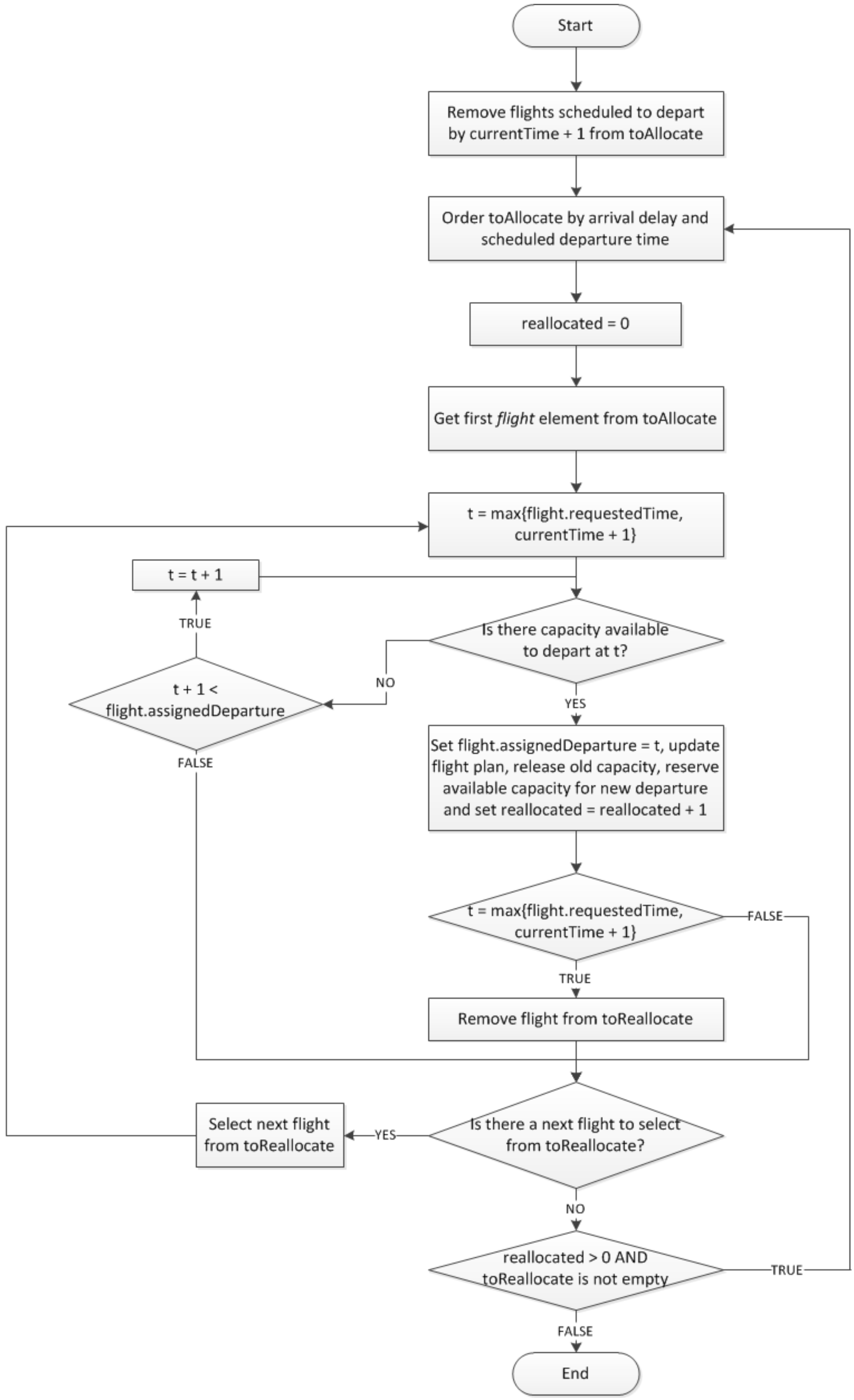

Figure 4.4: Reallocation of available capacity resources

4.2.3 Example

An example of the functioning of this algorithm is now provided. In this example, three flights, called f_1, f_2 and f_3, that are scheduled to depart from the same airport, are considered. Also, for the sake of simplicity, only operations taking place at the departure airport are considered. Capacity is only considered to be a critical factor here, with enough capacity available along the rest of the route of the considered flights. At time $t = 1$, the following information is available:

- f_1 has a departure time window [1, 3] and can depart at $t = 1$
- f_2 has a departure time window [1, 2], but its operating airline requests for a delayed departure at $t = 3$
- f_3 has a departure time window [1, 3], but its operating airline requests for a delayed departure at $t = 4$

Furthermore, after the execution of block (a) of the algorithm, f_2 is assigned an updated departure time at $t = 4$, while f_3 is assigned an updated departure time at t=6. Supposing that the non-collaborative approach is followed, the departure of f_1 at $t = 1$ causes the release of the capacity that it is not going to use. Specifically, there is one more unit of capacity available at $t = 2$ and $t = 3$ for flights to depart. Block (b) of the algorithm can now be executed, resulting in the following operations:

- sorting of flights: f_3 precedes f_2;
- analysis of f_3: capacity to reschedule the departure at an earlier time is not found;
- analysis of f_2: the flight can depart at $t = 3$, it is removed from the *toAllocate* set, and capacity to depart at $t = 4$ is released;
- the algorithm managed to reduce some delay during the last execution ($reallocated = 1$), so it is executed again;
- sorting of flights: f_3 is the only flight left;
- analysis of f_3: the flight can depart at $t = 4$, it is removed from the *toAllocate* set, and capacity to depart at $t = 6$ is released;
- the algorithm managed to reduce some delay during the last execution ($reallocated = 1$), but no flights are left in *toAllocate*, so the algorithm is terminated.

This example results in the complete removal of the increased additional delay assigned to flights: both f_2 and f_3 are able to depart at the new departure times requested by airlines. The results from computational tests using the discussed algorithm are discussed in the next section.

4.3 RTFRTW Computational Results

In this section, the results of the computational experiments performed to assess the practical viability of the proposed approach are presented. For each experiment, three steps are executed. First, time windows are identified. Second, reassignment requests are received, assigning increased additional delays (block (a) of the algorithm) when needed. Finally, the increased additional delays (block (b) of the algorithm) are reduced. Two different instance sets are considered. The first instance set consists of simulated instances, and the corresponding results are discussed in §4.3.1. The second set of instances is based on real European air traffic data, and results from the computational experience on this set of instances are discussed in §4.3.2. Both the non-collaborative and collaborative approaches are considered and evaluated.

4.3.1 Simulated instances

The computational experiments on simulated instances are based on the results of the ATFMTW model with a near-optimal model presented in §3.8.2. The actual departure times of flights, which can generate requests for new resource allocations to depart, are defined in accordance to a realistic probability distribution of delays based on data derived from EUROCONTROL's Demand Data Repository (DDR). An analysis on this data source was performed by Corolli, Castelli and Lulli [82]. In particular, based on real data from December 19th 2011, the additional delay is computed as the difference between the radar off block time and the last filled flight plan off block time. These figures are available in the DDR m3 and m1 .s06 files, respectively. The distribution of additional delay is depicted in Figure 4.5. The percentage of departures that are executed within the assigned time windows after the application of additional delay is determined on a set of 200 simulations per instance and per capacity level. Figure 4.6 shows that this percentage, as expected, increases with the available capacity, but ranging between very close values. In fact, it varies from 73.3% when the bad weather front is the strongest to 74.1% when no bad weather front is present.

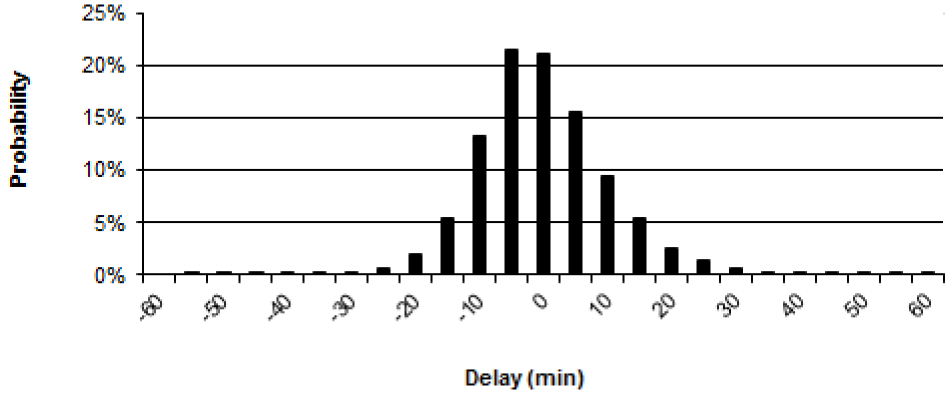

Figure 4.5: PROBABILITY DISTRIBUTION OF ADDITIONAL DELAYS

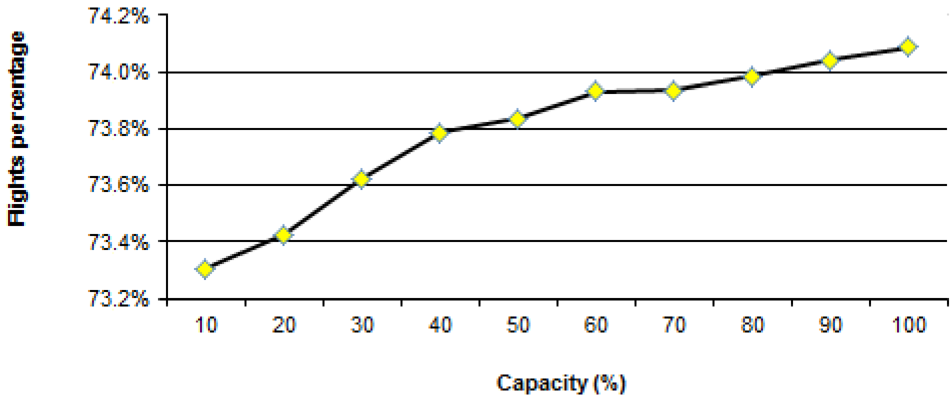

Figure 4.6: PERCENTAGE OF RESPECTED TIME WINDOWS

Every flight not respecting its departure time window may be subject to an increased additional delay. The total amount of increased additional delay in the network is strongly influenced by the attitude of airlines to share their information. In fact, assuming that five time instants (25 minutes) before the scheduled departure of a flight the corresponding airline is in the position to know whether it can respect or not its departure time window, the figures of increased additional delay from the simulations on the set of simulated instances are shown in Figure 4.7. Here, it is possible to observe that the increased additional delay initially assigned – i.e., after the execution of block (a) of the algorithm – is much higher using the non-collaborative approach. For example, in the case with capacity reduced to 10% of the nominal, the total increased additional delay is equal to 268" and 338" per flight with the collaborative and non-collaborative approach,

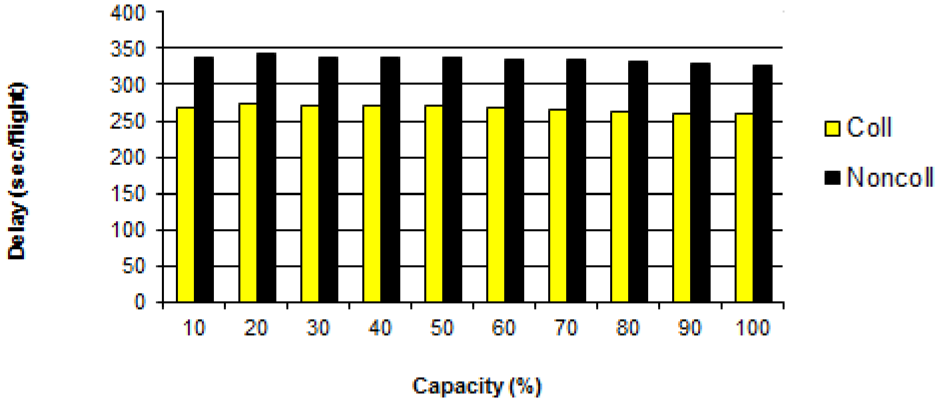

Figure 4.7: Initial increased additional delay

respectively. These figures correspond to a total of 28,910 and 36,497 minutes of delay on the whole system for the two different approaches. Among all the different capacity reduction levels, the increased additional delay is from 24.3% to 26.6% higher following the non-collaborative approach.

Once the increased additional delay is assigned to all flights, it can be reduced through the delay recovery algorithm from block (b). Figure 4.8 shows that most of the increased additional delay can be eliminated. The increased additional delay is reduced by 99.8% for instances with no bad weather fronts using both the collaborative and non-collaborative approaches. The smallest gain is obtained for the instances with the heaviest bad weather fronts, where 97.6% and 97.8% of the increased additional delay is removed with the two different approaches. The quantity of remaining increased additional delay is very much dependent on the available capacity: using the collaborative approach, it ranges from 6.4" to 0.4" per flight, i.e., from 696' to 46' on the whole system. Using the non-collaborative approach, on the other hand, the remaining increased additional delay ranges from 7.3" to 0.8" per flight, i.e., from 785' to 85' on the whole system. The difference between the two attitudes is small, as it ranges between 0.8" and 0.4" per flight, i.e., between 89' and 39' on the whole system. This result seems to justify the use of the non-collaborative approach as the price to pay, in terms of absolute added delay, is very low. Furthermore, a non-collaborative approach allows airline operators to deal with possible delays that may be added to a flight between the time when the exact time of departure is formulated and the departure itself, as flights maintain their time windows until they actually depart.

The percentage of respected time windows and the limited increased additional delay obtained using the non-collaborative approach show that it is possible to efficiently

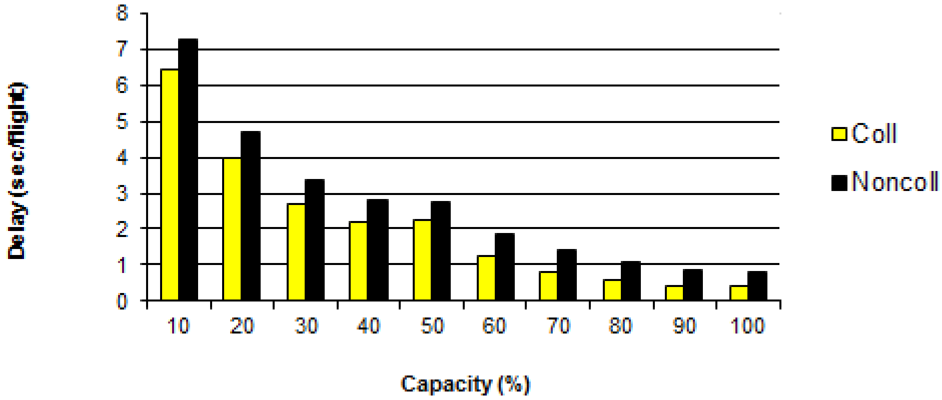

Figure 4.8: Final increased additional delay

operate flights using time windows even in uncertain situations where flights may not respect them. This is very important for their possible application to real flights, where the ability of dealing with uncertainty is fundamental. Furthermore, the average runtime of the delay reduction algorithm, i.e., block (b), was 49ms and 61ms using the collaborative and non-collaborative approach, respectively. These estimates refer to executions of the algorithm that considered for delay reduction a minimum of 100 flights. Therefore, the practical impact of the computation times to execute the proposed algorithm is negligible, making it usable for a real-time application.

4.3.2 Real data instances

A set of experimental computations based on real European air traffic data is also considered. EUROCONTROL DDR data from May 23rd to 29th 2011 were used as input. This is a whole week, from Monday to Sunday, with significant differences in the number of flights flown: Friday is the busiest day, and then comes Thursday, the other weekdays and finally the two weekend days. All flights scheduled to depart from European airports in the time interval between 12pm to 6pm every day are considered. Time windows are applied to last filled flight plans, while departure times specified in radar data are used as the requested flight times. Due to the lack of real data on sector capacity constraints, only airport capacity constraints are enforced. Since most of the congestion typically arises at airports, this is not a big drawback for the computational experiments. The figures of airport capacities are also taken from DDR data.

The results of the application of the delay reduction algorithm to real flight data are reported in Table 4.1. The table has the following structure. Column 2 shows the number of flights considered for each day. Additional delays are considered as the difference between the radar off block time and the last filled flight plan off block time, as in the simulated experimentations. Column 3 shows the average additional delay per flight. Column 4 shows the percentage of departure time windows that are respected after the introduction of the additional delay. Notice that they are inversely proportional to the average additional delay (correlation coefficient = -0.99). Furthermore, in highly trafficked days (May 26th and 27th), this percentage is significantly lower than in the case of simulated data. Columns 5 to 8 present the average amount of increased additional delay before and after the execution of the delay recovery algorithm using the collaborative and non-collaborative approaches. Again, in cases of high traffic the system performances are significantly worse. Initial increased additional delays are very similar under the two approaches. On the other hand, the delay recovery algorithm allows to reduce the increased additional delay to a larger extent using the collaborative approach. The last two columns of the table show that airline collaboration allows to save at least 14% of the increased additional delay.

These figures are significantly different from those of the tests on simulated instances. Using real data, in fact, increased additional delay is not negligible, therefore saving some of this delay can lead to tangible improvements. The great variability of results from day to day highlights the difficulty of realistically simulating an air traffic system. Given the non-negligible increased additional delays that can be saved by adopting a collaborative approach among airlines, its adoption can be desirable to improve the performances of a system that uses time windows. In fact, the collaborative approach allows to save between 7" and 26" of delay per flight in the considered real life cases.

Table 4.1: Results on instances from real data

				Increased Additional Delay					
		Add.	Resp.	Collaborative		Noncollaborative			
Day	Flights	delay	TW (%)	Initial	Final	Initial	Final	Time	%
29	7991	1’23"	76.7%	13’44"	46"	14’20"	57"	11"	19.8%
24	7649	1’29"	76.4%	11’23"	42"	11’40"	50"	8"	16.9%
25	7952	1’40"	75.5%	9’10"	44"	9’36"	51"	7"	14.1%
23	8029	1’59"	74.4%	14’59"	38"	15’12"	45"	7"	15.1%
28	6344	4’11"	70.6%	11’40"	36"	16’56"	1’02"	26"	42.2%
26	8028	5’28"	67.3%	15’30"	50"	15’28"	59"	9"	15.0%
27	8372	5’46"	65.0%	28’35"	1’17"	28’46"	1’31"	13"	14.5%

4.4 RTFRTW Conclusions

By defining as time window the interval of time between the maximum additional delay and the scheduled delay – possibly null, corresponding to the scheduled time – imposed by the Network Manager, this chapter evaluates the practical applicability of the time window concept in realistic scenarios. In particular, when a flight has a departure time window of, say, three time periods, this means that this flight occupies one unit of the airport capacity at each of these time instants. However, if the actual time of departure within this time window is known in advance, only one unit of the airport capacity can be reserved to the flight, thus allowing the remaining units to be used by other flights. Then, in accordance with the ability of airlines to predict their operations and their willingness to disclose them, two different approaches can be followed. The first is a collaborative approach, where airline operators always announce the exact departure time of a flight some time before its execution. The second is a non-collaborative approach, where airline operators do not announce delays within a flight's departure time window. Using the latter attitude, unused capacity is released by airline operators late, or may not be released at all, resulting in increased delays for other flights. This attitude however allows airline operators to deal with other unexpected delays that may occur between the time of the formulation of the updated time of departure and the departure itself.

Relying on simulated air traffic data, the reallocation of unused capacity is shown to be just slightly more effective, i.e., the overall network delay is lower, when airlines collaborate and disclose their exact departure time in advance. Hence, it might be possible to conclude that the non-collaborative approach is justifiable since it produces only a small deterioration of the network performances while granting higher flexibility to all flights. However, results based on real air traffic data of the European airspace suggest that the network average delay per flight is indeed significantly lower when airlines collaborate, with collaboration resulting in a non-negligible positive effect on delays. Furthermore, both simulated and real flight data show that, in non-congested situations, time windows can be respected by more than 73% of flights, which is close to three out of four flights. Time windows can therefore allow a good share of flights to perform their operations without requiring the network manager any rescheduling activity.

Assignment of capacity resources to flights rather than to airlines, however, may be practically problematic. While this approach allows for the highest system flexibility, the actual willingness of airlines to collaborate is doubtful. In fact, airlines are typically not willing to release resources that may be used by their competitors, reducing their costs.

This practical difficulty is not faced in this work. Penalties may be applied to airlines not releasing unused resources, however such an approach is not likely to be accepted by airlines. A future development of this work should therefore study the assignment of capacity resources to airlines rather than to flights, to address this issue. A closely related issue is the tradeoff between the fairness of delay allocation among airlines and the overall network delay, which should also be studied. Another future development will be devoted to refining and updating the analysis of the real instances, including sector capacity constraints to represent the complete real air traffic system. Finally, different priority rules to be used in the delay recovery algorithm will be evaluated, to better reflect RBS mechanisms.

Chapter 5
CONCLUSIONS

In this thesis, scheduling problems at different decision making phases from the field of air traffic are faced. Solving such problems is very important in practice, as delays in air traffic management are very expensive for airlines. A study of the University of Westminster [7] estimated the cost of ATFM delays to 83€ per minute, including direct costs, the network effect, and the estimated costs for airlines to retain passenger loyalty. Part of these costs results from non-deterministic events, such as bad weather fronts that cause a reduction of available capacity or last minute problems involving the staff, passengers, or aircraft of a flight. Considering such uncertainty can lead to important monetary benefits for airlines. This work is a first attempt to coherently face air traffic scheduling problems at different decision phases within a unique framework.

The decision making process that involves flight schedules and flight plans spans over a wide time horizon. Schedules are first defined months before the time of operation of flights, translating airline requests into schedules. These decisions are made at the strategic decision making phase, when uncertainty on future weather conditions is extreme. In fact, for generating capacity availability scenarios, it is in general only possible to use historical data on weather or capacity availability, or seasonal forecasts, which are very unprecise. This results in the definition of a huge number of scenarios. The problem of defining the first flight schedules considering such uncertainty at this phase is called Time Slot Allocation Problem under Uncertainty. This problem is faced in chapter 2, where a two-stage stochastic programming model with two alternative formulations is proposed to solve it. Due to the possible extreme number of scenarios at this phase, Sample Average

Approximation is used to solve problem instances. Results on instances representative of different airport networks in Europe indicate that considering uncertainty and solving the problem over the whole network can lead to a great reduction of delays for airlines. In an instance representative of a network of airports in France, the improvement over a schedule that does not consider uncertainty on capacity is equal to 45%, showing the great potential of this approach. Larger airport networks will be analyzed in future developments of this research.

On the day of operation of flights, a few hours before their execution, complete flight plans are defined. Defining flight plans at this phase is known as Air Traffic Flow Management. This problem is faced in chapter 3. Two different types of uncertainty are considered. First, the uncertainty on capacity availability is considered. Then, the implicit uncertainty on the departure time of flights is taken into account. Uncertainty on capacity availability at this phase is limited, as scenarios can be derived from weather forecasts, which are in general quite accurate for the same day. A two-stage stochastic programming model with two alternative formulations is proposed, as well as an ad-hoc heuristic to solve problem instances within short computation times. The execution of the proposed stochastic programming model shows that stochastic programming can provide airlines with relevant ATFM delay cost reductions when bad weather affects a large portion of the airspace. In particular, among the studied instances, it was possible to save over 14% of ATFM delay cost in a case considering bad weather affecting the whole network.

The implicit uncertainty on the departure time of flights is managed using time windows which determine the time intervals flights can use to execute their operations without incurring in any additional delay. The narrower a time window, the more critical the execution of the related flight operation. Two deterministic models to define time windows are proposed. The first model, called optimal model, is inefficient in terms of computation times, as it makes a joint decision on the schedule and the time windows. An approach that can be used in practice is provided by the near-optimal model, which provides maximum time windows for a given schedule in only 40" on average for large-scale instances. Furthermore, different criteria to define time windows have been compared through simulation, since reserving capacity at each instant of a time window results in declaring too many flights as critical. This analysis highlights that at least 99.8% of airport/sector constraints are respected in practice when time windows are defined proportionally, i.e., the amount of capacity reserved at each instant of a time window is inversely proportional to the width of the time window. This suggests that time windows can be a viable tool to deal with the implicit uncertainty on the departure time of flights without having to declare too many flights as critical.

Finally, when operations take place, i.e., at the operational phase, some adjustments may need to be done to schedules. In fact, some flights may not be able to meet the assigned departure time windows, resulting in their request for new capacity resources in order to depart at a later time. Capacity resources can be retrieved from departing flights and reassigned to flights that do not respect time windows. This problem is called Real Time Flight Rescheduling with Time Windows, and is discussed in chapter 4. Since the problem should be solved in real time, a rule-based algorithm based on RBS is used to perform resource allocation according to some predefined precedence rules. Two different airline behaviors are compared: one that is collaborative, and one that is non-collaborative. When airlines collaborate, computational tests on real traffic data indicate that additional delays can be reduced by at least 14%. This can stimulate airlines to adopt a collaborative approach, which is in line with the Collaborative Decision Making goals.

Future research will focus on the further development of each single work, maintaining the coherence between them to improve the whole framework. Also, some possible conjunctions between different solutions may be studied. For example, the uncertainty on capacity availability and the implicit uncertainty on the departure time of flights may be considered together in a single model. Implementing time windows in Stochastic Air Traffic Flow Management, however, is not a trivial step, due to the complexity of reserving stochastic capacity and taking appropriate recourse actions. The works presented in this thesis, however, already indicate a wide set of improvements that can be made in managing air traffic operations over a wide time horizon, which can result in relevant monetary benefits for airlines and other stakeholders.

ACKNOWLEDGEMENTS

The most important acknowledgements go to those thanks to whom my experience as a Ph.D. student was possible. First of all my supervisor, Prof. Guglielmo Lulli, for his contribution to my development and his patience over the years. Thanks to my tutor, Prof. Marco Antoniotti, who has always provided me with good advice, and to the coordinator of the Ph.D. school, Prof. Stefania Bandini, especially for stimulating me to improve my presentations. Thanks to the professors and staff of the whole department (Dipartimento di Informatica Sistemistica e Comunicazione) that hosted me for 3 years, the University of Studies of Milan-Bicocca, and the Italian Government for supporting me for three years with a scholarship.

A special acknowledgement goes to Prof. Lewis Ntaimo from Texas A&M University for the great contribution to my development and for the hospitality provided during my visiting period at his University, along with all the Department of Industrial and Systems Engineering of Texas A&M University. Another special acknowledgement goes to Prof. Lorenzo Castelli from the University of Studies of Trieste, who greatly contributed to my development and introduced me to research in 2009.

Important contributions were also provided by the other Ph.D. students I have interacted and/or worked with during these three years, both at the University of Studies of Milan-Bicocca and at Texas A&M University. Acknowledgement must also be given to two student organizations that have differently contributed to my development. First, acknowledgement goes to the international students' association "Electrical Engineering STudents' European assoCiation" (EESTEC), for the great contribution to my skills in team building and public speaking, and for introducing me to many different European cultures. Also, I wish to thank the "Study Turkish" student organization at Texas A&M,

for providing me with a free and good quality Turkish language course, and for involving me in the celebrations of the local Turkish community.

This work would have never been done without the great support from my family, who has supported me regardless of where I was, and to my friends from all over the world, who all supported me in different, but always important, ways. Among my friends, a special acknowledgement goes to “The Walruses”.

Finally, I wish to thank those people who supported me that are not among us anymore.

BIBLIOGRAPHY

[1] S. Hartmann and D. Briskorn. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1–14, 2010.

[2] EUROCONTROL Performance Review Commission. Performance review report 2012. May 2013. URL `http://www.eurocontrol.int/documents/performance-review-report-prr-2012`.

[3] EUROCONTROL STATFOR. Forecast of Annual Number of IFR Flights (2013 – 2019). February 2013. URL `http://www.eurocontrol.int/documents/seven-year-flights-forecast-2013-2019`.

[4] EUROCONTROL. Challenges of Growth 2013. June 2013. URL `http://www.eurocontrol.int/articles/challenges-growth`.

[5] SESAR Joing Undertaking. Long-Term and Innovative Research – WP-E Thematic Programme. May 2012. URL `http://www.sesarju.eu/sites/default/files/documents/workpackages/wpe/12-120610_WP-E_Thematic_Programme_v3.0.pdf`.

[6] Airports Council International. ACI Economics Survey 2011.

[7] Department of Transport Studies University of Westminster London. European airline delay cost reference values. March 2011. URL `http://www.eurocontrol.int/sites/default/files/content/documents/single-sky/pru/publications/other/european-airline-delay-cost-reference-values-final-report-v3.2.pdf`.

[8] Department of Transport Studies University of Westminster London. European airline delay cost reference values. March 2011. URL `http://www.eurocontrol.int/sites/default/files/content/documents/single-sky/pru/publications/other/european-airline-delay-cost-reference-values-final-report-v3.2.pdf`.

[9] M.O. Ball, C. Barnhart, G. Nemhauser, and A.R. Odoni. Air transportation: irregular operations and control. In *Handbooks of Operations Research and Management*, 2006.

[10] J.R. Birge and F.V. Louveaux. *Introduction to Stochastic Programming.* Springer, New York, 1997.

[11] J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[12] R. Van Slyke and R.B. Wets. L-Shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4): 638–663, July 1969.

[13] G. Infanger. *Planning under uncertainty: solving large scale stochastic linear programs.* Boyd and Fraser Publishing Company.

[14] J.L. Higle and S. Sen. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research*, 16(3):650–669, August 1991.

[15] J.L. Higle and S. Sen. *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming.* Kluwer Academic Publishers, February 1996.

[16] R. de Neufville and A.R. Odoni. *Airport Systems: Planning, Design and Management.* McGraw-Hill, October 2003.

[17] J.K. Brueckner. Internalisation of airport congestion. *Journal of Air Transport Management*, 8(3):141–147, 2002.

[18] T.P. Fan. *Airport Demand Management and Congestion Pricing.* Ph.D. Dissertation, Massachusetts Institute of Technology, Center for Transportation and Logistics, Cambridge (MA), 2002.

[19] M. Raffarin. Congestion in european airspace: A pricing solution? *Journal of Transport Economics and Policy*, 38(1):109–125, January 2004.

[20] International Air Transport Association (IATA). *Worldwide Slot Guidelines, 2nd Edition*, January 2012.

[21] A.I. Czerny, P. Forsyth, D. Gillen, and H.-M. Niemeier, editors. *Airport slots: international experiences and options for reform.* Ashgate, February 2008.

[22] K.G. Zografos, Y. Salouras, and M.A. Madas. Dealing with the efficient allocation of scarce resources at congested airports. *Transportation Research Part C: Emerging Technologies*, 21(1):244–256, April 2012.

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

[24] M.O. Ball, G.L. Donohue, and K. Hoffman. Auctions for the safe, effcient and equitable allocation of airspace system resources. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*, chapter 20. MIT Press, 2005. doi: 10.7551/mitpress/9780262033428.003.0021.

[25] A.N. Kleit and B.H. Kobayashi. Market failure or market efficiency? evidence on airport slot usage. *Research in Transportation Economics*, 4:1–32, 1996.

[26] H. Fukui. An empirical analysis of slot trading in the Unites States. *Transportation Research Part B: Methodological*, 44(3):330–357, March 2010.

[27] E.T. Verhoef. Congestion pricing, slot sales and slot trading in aviation. *Transportation Research Part B: Methodological*, 44(3):320–329, March 2010.

[28] Mott MacDonald Group. Study on the impact of the introduction of secondary trading at community airports. Technical report, November 2006. URL `http://www.euaca.org/up/files/DocsEUROPE/EU_REGULATION_95_93_AS_AMENDED_Slot_Regulation/2006_slots_final_report.pdf_211108_054651.pdf`.

[29] D. Starkie. Allocating airport slots: a role for the market? *Journal of Air Transport Management*, 4(2):111–116, April 1998.

[30] P. Pellegrini, L. Castelli, and R. Pesenti. Secondary trading of airport slots as a combinatorial exchange. *Transportation Research Part E: Logistics and Transportation Review*, 48(5):1009–1022, Septebmer 2012.

[31] D. Kösters. Airport scheduling performance – an approach to evaluate the airport scheduling process by using scheduled delays as quality criterion. In *Proceedings of the Air Transport Research Society (ATRS) Annual World Conference*, 6 2007.

[32] P. Pellegrini, L. Castelli, and R. Pesenti. Metaheuristic algorithms for the simultaneous slot allocation problem. *IET Intelligent Transport Systems*, 6(4):453–462, December 2012. doi: 10.1049/iet-its.2011.0179.

[33] A.M. Churchill, D.J. Lovell, A. Mukherjee, and M.O.Ball. Determining the number of airport arrival slots. *Transportation Science*, in press, 2013.

[34] K.G. Zografos, M.A. Stamatopoulos, and A.R. Odoni. An analytical model for runway system capacity analysis. In *Proceedings of the 8th International Federation of Automatic Control (IFAC) Symposium*, volume 1, pages 29–34, June 1997.

[35] G. Andreatta, L. Brunetta, A.R. Odoni, L. Righi, M.A. Stamatopoulos, and K.G. Zografos. A set of approximate and compatible models for airport strategic planning on airside and on landside. In *Proceedings of the 2nd USA/Europe Air Traffic Management R&D Seminar*, December 1998.

[36] L. Brunetta, L. Righi, and G. Andreatta. An operations research model for the evaluation of an airport terminal: SLAM (simple landside aggregate model). *Journal of air traffic management*, (5):161–175, 1999. doi: 10.1016/S0969-6997(99)00010-1.

[37] G. Andreatta, L. Brunetta, and L. Righi. Evaluating terminal management performances using slam: The case of athens international airport. *Computers & Operations Research*, 34(6):1532–1550, June 2007. doi: 10.1016/j.cor.2005.07.024.

[38] L. Brunetta, G. Andreatta, L. Righi, and G. Romanin-Jacur. Evaluating terminal management performances: the case of AIA terminal and the use of E-SLAM. In *Proceedings of Odysseus 2003*, May 2003.

[39] A. Ruszczyński and A. Shapiro. *Stochastic Programming. Handbooks in Operations Research and Management Science*, volume 10. Elsevier. ISBN 978-0-444-50854-6.

[40] ILOG. *IBM ILOG CPLEX 12.1 Callable Library C API Reference Manual.* 2009.

[41] Airports Council International Europe. ACI EUROPE position on the allocation of airport slots. December 2011. URL `https://www.aci-europe.org/component/downloads/downloads/3110.html`.

[42] G. Andreatta, L. Brunetta, and G. Guastalla. The flow management problem: recent computational algorithms. *Control Engineering Practice*, 6(6):727–733, June 1998. doi: 10.1016/S0967-0661(98)00078-1.

[43] G. Andreatta, L. Brunetta, and G. Guastalla. From ground holding to free flight: An exact approach. *Transportation Science*, 34(4):394–401, November 2000. doi: 10.1287/trsc.34.4.394.12318.

[44] A. Agustín, A. Alonso-Ayuso, L.F. Escudero, and C. Pizarro. Mathematical optimization models for air traffic flow management: a review. In *Combinatorial Optimization in Practice*, pages 141–184. Studia Informatica Universalis, A. Bui, I. Tseveendorkj (Eds.), February 2010.

[45] O. Richetta and A.R. Odoni. Dynamic solution to the ground-holding problem in air traffic control. *Transportation Research Part A: Policy and Practice*, 28(3):167–185, May 1994.

[46] A. Alonso, L.F. Escudero, and M.T. Ortuno. A stochastic 0-1 program based approach for the air traffic flow management problem. *European Journal of Operational Research*, 120(1):47–62, 2000.

[47] D. Bertsimas and S. Stock Patterson. The air traffic flow management problem with enroute capacities. *Operations Research*, 46(3):406–422, May/June 1998. doi: 10.1287/opre.46.3.406.

[48] A. Nilim and L. El Ghaoui. Algorithms for air traffic flow management under stochastic environments. In *Proceedings of the 2004 American Control Conference*, number 4, pages 3429–3434, June/July 2004.

[49] A. Mukherjee and M. Hansen. Dynamic stochastic optimization model for air traffic flow management with en route and airport capacity constraints. In *Proceedings of the 6th USA-Europe Air Traffic Management Research and Development Conference*, June 2005.

[50] Y-H. Chang. *Stochatic programming approaches to air traffic flow management under uncertainty of weather.* Ph.D. Dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology, December 2010.

[51] G. Andreatta, P. Dell'Olmo, and G. Lulli. An aggregate stochastic programming model for air traffic flow management. *European Journal of Operational Research*, 215(3):697–704, December 2011.

[52] A. Agustín, A. Alonso-Ayuso, L.F. Escudero, and C. Pizarro. On air traffic flow management with rerouting. part II: Stochastic case. *European Journal of Operational Research*, 219(1):167–177, May 2012.

[53] D. Bertsimas and S. Gupta. Multistage air traffic flow management under capacity uncertainty: A robust and adaptive optimization approach. *TSL Workshop*, June 2011.

[54] G. Lulli and A.R. Odoni. The european air traffic flow management problem. *Transportation Science*, 41(4):431–443, November 2007. doi: 10.1287/trsc.1070.0214.

[55] E.A. Boyd. Fenchel cutting planes for integer programs. *Operations Research*, 42 (1):53–64, January/February 1994.

[56] SESAR Consortium. Milestone Deliverable D3: ATM Target Concept. September 2007. URL `http://www.eurocontrol.int/sesar/gallery/content/public/docs/DLM-0612-001-02-00.pdf`.

[57] A.R. Odoni. The flow management problem in air traffic control. In A.R. Odoni, L. Bianco, and G. Szegö, editors, *Flow Control of Congested Networks*, pages 269–288. Springer, 1987.

[58] M. Helme. Reducing air traffic delay in a space-time network. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 236–242, October 1992. doi: 10.1109/ICSMC.1992.271770.

[59] K.S. Lindsay, E.A. Boyd, and R. Burlingame. Traffic flow management modeling with the time assignment model. *Air Traffic Control Quarterly*, 1(3):255–276, 1993.

[60] D. Bertsimas and S. Stock Patterson. The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach. *Transportation Science*, 34 (3):239–255, August 2000. doi: 10.1287/trsc.34.3.239.12300.

[61] A.J. Alonso and L.F. Escudero. 0-1 equivalent model representations for air traffic management. *Top*, 6(1):19–36, 1998. doi: 10.1007/BF02564796.

[62] D. Sun and A.M. Bayen. Multicommodity eulerianâĂŞlagrangian large-capacity cell transmission model for en route traffic. *Journal of Guidance, Control, and Dynamics*, 31(3):616–628, 2008. doi: 10.2514/1.31717.

[63] T. Myers and D. Kierstead. Network model to address capacity/demand imbalances in the national airspace system. In *American Institute of Aeronautics and Astronautics (AIAA) Guidance, Navigation and Control Conference and Exhibit*, pages 1774–1798, 2008. ISBN 9781605608082.

[64] EUROCONTROL Central Flow Management Unit (CFMU). General & CFMU Systems. March 2010. URL `http://www.worldairops.com/EUR/docs/EUR_BasicCFMUHandbook_General_atWorldAirOps.com.pdf`.

[65] D. Bertsimas, G. Lulli, and A.R. Odoni. An integer optimization approach to large-scale air traffic flow management. *Operations Research*, 59(1):211–227, January 2011. doi: 10.1287/opre.1100.0899.

[66] A.M. Churchill, D.J. Lovell, and M.O. Ball. Evaluating a new formulation for large-scale traffic flow management. In *Proceedings of the 8th USA/Europe Air Traffic Management R&D Seminar*, 2009.

[67] A. Agustín, A. Alonso-Ayuso, L.F. Escudero, and C. Pizarro. On air traffic flow management with rerouting. part I: Deterministic case. *European Journal of Operational Research*, 219(1):156–166, May 2012.

[68] D. Sun, A. Clinet, and A.M. Bayen. A dual decomposition method for sector capacity constrained traffic flow optimization. *Transportation Research Part B: Methodological*, 45(6):880–902, July 2011.

[69] C. Brinton, S. Atkins, L. Cook, S. Lent, and T. Prevost. Ration by schedule for airport arrival and departure planning and scheduling. In *Proceedings of the 2010 Integrated Communications Navigation and Surveillance (ICNS) Conference*, May 2010. doi: 10.1109/ICNSURV.2010.5503239.

[70] M.O. Ball, C.-Y. Chen, R. Hoffman, and T. Vossen. Collaborative decision making in air traffic management: Current and future research directions. In *New Concepts and Methods in Air Traffic Management*, Transportation Analysis, pages 17–30. Springer, Bianco-Dell'Olmo-Odoni (Eds.), 2001. doi: 10.1007/978-3-662-04632-6_2.

[71] M.C. Wambsganss. Collaborative decision making air traffic management. In *New Concepts and Methods in Air Traffic Management*, Transportation Analysis, pages 1–15. Springer, Bianco-Dell'Olmo-Odoni (Eds.), 2001. doi: 10.1007/978-3-662-04632-6_1.

[72] M.O. Ball, R. Hoffman, W. Hall, and A. Muharremoglu. *Collaborative decision making in air traffic management: A preliminary assessment.* Technical Report RR-99-3, NEXTOR, The National Center of Excellence for Aviation Operations Research, 1999.

[73] T. Vossen. *Fair Allocation Concepts in Air Traffic Management.* Ph.D. Dissertation, University of Maryland at College Park, July 2002.

[74] M.O. Ball and G. Lulli. Ground delay programs: Optimizing over the included flight set based on distance. *Air Traffic Control Quarterly*, 12(1):1–25, 2004. ISSN 1064-3818.

[75] C. Brinton, C. Provan, S. Lent, T. Prevost, and S. Passmore. Collaborative departure queue management. In *Proceedings of the 9th USA/Europe Air Traffic Management Research and Development Seminar*, 2011.

[76] J.C. Jones, D.J. Lovell, and M.O. Ball. Algorithms for dynamic resequencing of en route flights to relieve terminal congestion. In *Proceedings of the 5th International Conference on Research in Air Transportation (ICRAT)*, May 2012.

[77] T. Vossen and M.O. Ball. Optimization and mediated bartering models for ground delay programs. *Naval Research Logistics (NRL)*, 53(1):75–90, 2006. doi: 10.1002/nav.20123.

[78] E.J. Cholankeril. *Evaluation of an Auction Mechanism for Allocating Airport Arrival Slots.* Ph.D. Dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2003.

[79] H. Balakrishnan. Techniques for reallocating airport resources during adverse weather. In *46th IEEE Conference on Decision and Control*, pages 2949–2956, December 2007. doi: 10.1109/CDC.2007.4434783.

[80] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, March 1961.

[81] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasining about a Highly Connected World*. Cambridge University Press, 2010.

[82] L. Corolli, L. Castelli, and G. Lulli. The impact of uncertain departure delays on flight flexibility. In *Proceedings of the 5th International Conference on Research in Air Transportation (ICRAT)*, May 2012.