Air Traffic Flow Management at Airports: A Unified Optimization Approach

by

Michael Joseph Frankovich

B.A., University of Auckland (2008)
B.E., University of Auckland (2008)

Submitted to the Sloan School of Management
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2012

Author . . . .
Sloan School of Management
August 17, 2012

Certified by . . . .
Dimitris J. Bertsimas
Boeing Leaders for Global Operations Professor
Co-director, Operations Research Center
Thesis Supervisor

Accepted by . . . .
Patrick Jaillet
Dugald C. Jackson Professor
Co-director, Operations Research Center

Air Traffic Flow Management at Airports: A Unified Optimization Approach

by

Michael Joseph Frankovich

Submitted to the Sloan School of Management
on August 17, 2012, in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy in Operations Research

Abstract

The cost of air traffic delays is well documented, and furthermore, it is known that the significant proportion of delays is incurred at airports. Much of the air traffic flow management literature focuses on traffic flows between airports in a network, and when studies have focused on optimizing airport operations, they have focused largely on a single aspect at a time.

In this thesis, we fill an important gap in the literature by proposing *unified* approaches, on both strategic and tactical levels, to optimizing the traffic flowing through an airport. In particular, we consider the entirety of key problems faced at an airport: a) selecting a runway configuration sequence; b) determining the balance of arrivals and departures to be served; c) assigning flights to runways and determining their sequence; d) determining the gate-holding duration of departures and speed-control of arrivals; and e) routing flights to their assigned runway and onwards within the terminal area.

In the first part, we propose an optimization approach to solve in a unified manner the strategic problems (a) and (b) above, which are addressed manually today, despite their importance. We extend the model to consider a group of neighboring airports where operations at different airports impact each other due to shared airspace.

We then consider a more tactical, flight-by-flight, level of optimization, and present a novel approach to optimizing the entire *Airport Operations Optimization Problem*, made up of subproblems (a) – (e) above. Until present, these have been studied mainly in isolation, but we present a framework which is both unified and tractable, allowing the possibility of system-optimal solutions in a practical amount of time.

Finally, we extend the models to consider the key uncertainties in a practical implementation of our methodologies, using robust and stochastic optimization. Notable uncertainties are the availability of runways for use, and flights' earliest possible touchdown/takeoff times. We then analyze the inherent trade-off between robustness and optimality.

Computational experience using historic and manufactured datasets demonstrates that our approaches are computationally tractable in a practical sense, and could result in cost benefits of at least 10% over current practice.

Thesis Supervisor: Dimitris J. Bertsimas
Title: Boeing Leaders for Global Operations Professor
Co-director, Operations Research Center

Acknowledgments[1]

I would firstly like to thank my advisor Dimitris Bertsimas, without whom this would not have happened. You always had a clear direction in mind. I admire your energy and greatly appreciate everything you have done for me. It has been a privilege working with you.

A special thanks is also due to Amedeo Odoni, a font of knowledge of the airline industry – thank you for your conscientious feedback, your kindness and your support.

I would like to thank Mariya Ishutkina, Rich Jordan, Jim Kuchar, Tom Reynolds and Ngaire Underhill of MIT Lincoln Laboratory, as well as Professor Hamsa Balakrishnan for providing me with much of the data necessary to complete this thesis. I am also very grateful for the feedback and direction I received from many of the same, and also from Bill Moser and Mark Weber of MIT Lincoln Laboratory, Professor Cindy Barnhart, and fellow students Ross Anderson, Chaithanya Bandi, Shubham Gupta, and Phil Keller.

Thank you to all those who have made the ORC such a special place to be a part of over these past few years. To Andrew and Laura – thank you for all the work you put in for us – you really do go beyond what's expected to make our lives easier, and it is much appreciated. To co-directors Cindy, Dimitris, and Patrick Jaillet, thank you for putting in your time and energy. Most of all, thank you to my fellow students and friends, I look back with great fondness upon the times we have shared.

To my Mum and Dad, I cannot adequately express my gratitude to you. Thank you for your love and support which has always been with me. Thank you to the rest of my family and friends for always being there, especially my wonderful sisters. Finally, to Andrea, thank you for your unwavering love and support, I couldn't imagine these years without you.

Cambridge, August 2012 *Michael Frankovich*

[1] This work was supported by MIT Lincoln Laboratory and the National Science Foundation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 2007, the cost of delays to US domestic flights on major airlines was estimated to be $8.3 billion, and the cost to passengers $16.7 billion [3]. Reducing theses delays and their associated costs represents a significant challenge for the struggling airline industry and in particular for the Federal Aviation Administration (FAA) – not only to increase profitability for airlines, many of which presently operate at a loss, but also to improve the experience for passengers. Furthermore, the importance of addressing these delays is emphasized by the fact that the total number of air traffic operations at combined FAA and contract towered airports is estimated to increase from 61.1 million in 2006 to 81.1 million by 2020 and 95.9 million by 2030 [19].

One way to reduce delays is to expand the air transportation infrastructure. This, however, is a very costly exercise in itself, and furthermore can take many years to successfully implement. Indeed, there is a consensus amongst experts in the airline industry that infrastructure development alone will not be enough to limit significant increases in delays above current levels [3]. As a result, there is a growing need to incorporate optimization into the air traffic flow management (ATFM) in order to minimize these delays. With reduced delays, also come reductions in emissions, as well as improved management of safety.

Much of the ATFM literature focuses on the traffic flows between airports in a network, with little focus on the operations at the airports themselves, in spite of the expansive and influential set of decisions which need to be made there. Furthermore, when previous studies have focused on optimizing airport operations, they have focused on a single aspect of the decisions made there at a time, for example runway sequencing or the gate-holding of departures. It is our belief that optimizing the traffic flowing *through* an airport, in all its complexity, is of critical importance; hence this is the focus of this thesis.

1.1 Background and Literature Review

1.1.1 Runway Configuration Management and Arrival/Departure Runway Balancing

The combination of runways which are active at any particular time at an airport is known as the “runway configuration” in which the airport operates at that time. The sequence of configurations selected by controllers greatly influences an airport’s capacity to serve demand for arrivals and departures. Runway configuration management (RCM) is concerned with determining a sequence of runway configurations to be used at an airport which will minimize delays (or maximize throughput, or some other related objective), while arrival/departure runway balancing (ADRB) is concerned with making a corresponding assignment of arrivals and departures to the active runways.

[23] and [24] proposed integer programs for the ADRB to minimize queueing by controlling the utilization of near-runway airspace and coordinating operations at arrival and departure fixes subject to a *fixed* airport capacity profile. There has been a distinct lack of work on the RCM problem, however, which we address in Chapter 2.

1.1.2 Runway Sequencing

The problem of sequencing flights at a single runway is known to be an application of the Traveling Repairman Problem (TRP), which is closely related to the Traveling Salesman Problem (TSP), differing in its objective function, being equal to the sum of the times each city (or flight) must wait before being arrived at (processed). This is because the minimum separation time required between each pair of flights depends on the type of each of the two flights, with different aircraft types producing different wake vortices, and these must clear sufficiently before another take-off/landing is safe to go ahead. In particular, the TRP problem here is a special case (B-TRPTW, to use the notation of [45]), having a fixed number of different types of aircraft (or a *bounded* number of locations at which calls can arrive, using the euclidean traveling repairman analogy), as well as *time windows*.

TRP and TSP have been studied in-depth, both more generally (see for example [45]) as well as in this application. Notably, [17], [34], [44], and [2] developed approaches which took advantage of the fairness principle that the optimal sequence should not differ too much from the first-come first-served (FCFS) sequence. Recently, [41] proposed a stochastic optimization approach to the runway scheduling problem.

1.1.3 Surface Management

There is also a substantial body of work on airport surface management and the gate-holding of departures. One objective here is to hold departures at the gate, with engines off, for as long as possible without delaying their take-off. In other words, delays in queue at runways or elsewhere on the taxiway system are transferred to delays at the gate. This results in less traffic on the surface, less fuel burn, and lower emissions. See for example [21], [35], [13] and [12]. Notably, [40] implemented a simple but effective "N-control" policy at BOS whereby the number of aircraft on the surface is restricted to reduce departure queue size, while also being large enough to ensure sequencing delays are not observed due to an insufficient pool of aircraft.

[30], [38] and [29] proposed approaches to the optimization of aircraft taxi routes, while [25], and more recently [15], merged the sequencing problem with the taxiing problem, recognizing as does this thesis the important interdependence between the two problems. However, their work ignores the important and complicating matter of runway configuration optimization.

1.1.4 Airport Optimization under Uncertainty

While there has been much work on the various airport optimization problems in the literature, there is a lack of work considering stochastic conditions. A notable example of such literature is [41], where a two-stage stochastic optimization based approach was developed to optimize the runway sequencing problem under uncertainty. In addition to [41], [26] also assessed the impact of uncertainty when implementing runway optimization policies. [32] considered the optimization of near-terminal air routes under uncertainty in weather conditions.

1.2 Contributions and Outline of Thesis

The broad goal of this thesis is to provide useful optimization models that address the key aspects of air traffic flow management at airports:

- The strategic use of airport capacity resources – selection of runway configurations and arrival/departure runway balancing;
- The tactical use of airport capacity resources – selection of runway configurations, runway assignment and sequencing, management of the airport surface and near-terminal airspace on a flight-by-flight level;
- Uncertainty in runway availability and in the earliest possible runway times of flights.

More detailed contributions, as well as the structure of the thesis are as follows:

In Chapter 2, we present a mixed integer programming (MIP) model to solve the problems of (i) selecting an airport's optimal sequence of runway configurations, and (ii) determining the optimal balance of arrivals and departures to be served at any moment. These problems, the Runway Configuration Management (RCM) Problem and the Arrival/Departure Runway Balancing (ADRB) Problem, respectively, are of critical importance in minimizing the delay of both in-flight and on-the-ground aircraft, along with their associated costs. We show that under mild assumptions on the time required to change between configurations, problem instances of realistic size can be solved within several seconds. Furthermore, as assumptions are relaxed, optimal solutions are still found within several minutes. Comparison with a sophisticated baseline heuristic reveals that in many cases the potential reduction in cost from using the method is significant, and could be expected to be of the order of at least 10%. Finally, we present an extension of the MIP model to solve these two problems for a group of airports in a metropolitan area such as New York ("metroplex"), where operations at each airport within the metroplex may have an impact on operations at some of the other airports due to limitations in shared airspace. Several of our results in this chapter have been reported in [6].

In Chapter 3, we present a novel integer optimization approach to optimize in a tractable and unified manner the *airport operations optimization problem* (AOOP). This includes solving the entirety of key air traffic flow management (ATFM) problems faced at an airport:

a) selecting a runway configuration sequence, i.e., determining which runways are open at which times and whether they will process arrivals and/or departures;

b) assigning flights to runways and determining the sequence in which flights are processed at each runway (i.e., when they take off or land);

c) determining the gate-holding duration of departures and speed-control of arrivals outside of the near-terminal airspace, if any;

d) routing flights to their assigned runway at the desired time and onwards within the terminal area and the near-terminal airspace.

There has been much work on these and related subproblems within the aviation and optimization communities, but as indicated in the above literature review, this work has focused mainly on a single subproblem at a time in isolation. In Chapter 3, however, we present what is to the best of our knowledge the first truly *unified and tractable optimization approach* to solve the overall ATFM problem (AOOP) at a single airport. That is, the first optimization approach which solves subproblems (a)

– (d) above together such that a (near-) system-optimal solution is attained within several minutes. The model is a general one – applicable to any airport, regardless of the runway, taxiway, or airspace design. This is a significant contribution due to both the size of the problem and the complexity of its subproblems, notably the runway sequencing subproblem. As a result of these characteristics, a naïve attempt to solve this overall problem would be far from computationally tractable, and it is only through our use of appropriate modeling that we have been able overcome this tractability challenge. Furthermore, solving the individual subproblems in isolation using the existing literature may lead to overall solutions which are grossly suboptimal, or indeed infeasible. Several of our results in this chapter have been reported in [5].

In Chapter 4, we consider the implementation of the optimization approach of Chapter 3 in a dynamic and uncertain real world environment. We extend the model using techniques from robust and stochastic optimization to consider the key uncertainties: i) in the earliest possible touchdown/takeoff times of flights, and ii) the availability of runways for use. We then perform a computational analysis using simulation to evaluate the effectiveness of the robust approach.

Finally, in the Concluding Remarks, we conclude with a summary and discussion.

Chapter 2

Strategic Airport Optimization

In this chapter, we present a mixed integer programming (MIP) model to solve the key air traffic flow management (ATFM) problems faced at an airport, at a strategic level. These are the problems of (i) selecting an airport's optimal sequence of runway configurations, and (ii) determining the optimal balance of arrivals and departures to be served at any moment. These problems, the Runway Configuration Management (RCM) Problem and the Arrival/Departure Runway Balancing (ADRB) Problem, respectively, are of critical importance in minimizing the delay of both in-flight and on-the-ground aircraft, along with their associated costs. We refer to them as "strategic" problems because they can be viewed as setting the operating mode of the airport, without specifying a flight-by-flight level of detail. We also present an extension of the model to solve these two problems for a group of airports operating in close proximity of one another (a "metroplex"), where operations at each airport may affect each other due to shared airspace.

2.1 Introduction

The combination of runways which are active at any particular time at an airport is known as the "runway configuration" in which the airport operates at that time. The sequence of configurations selected by controllers greatly influences an airport's capacity to serve demand for arrivals and departures. This chapter is concerned with

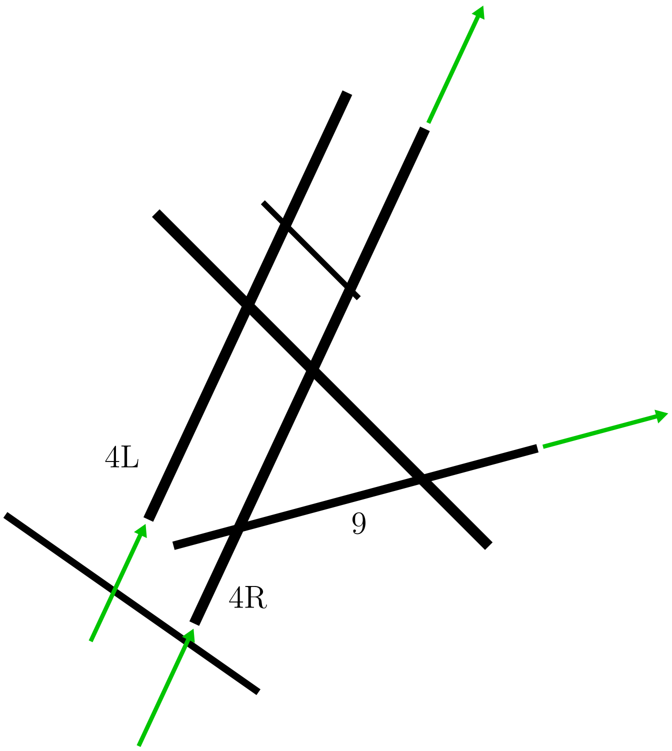

Figure 2-1: A runway configuration used at BOS, with the corresponding active runways indicated.

determining a sequence of runway configurations and the assignment of arrivals and departures to the active runways which, together, minimize the cost incurred at the airport due to delayed aircraft.

Figure 2-1 shows a schematic representation of Boston's Logan International Airport (BOS). The runway configuration shown in use in the figure consists of three active runways 4L, 4R and 9, with the others being idle. The direction of operations on each active runway is indicated, with two of the runways being used for arrivals only, while the third is used for departures only. In some configurations, it is possible that some runways, for example runway 4R in Figure 2-1, operate in "mixed mode," whereby both arrival and departure operations are permitted on a single runway, in the same direction of travel. For multi-runway airports, the number of possible configurations can be large. For instance, BOS typically employs more than 20 different configurations during a year, by making use of its runways in different ways.

The availability of certain configurations at most major airports may be restricted by the weather conditions prevailing at any particular time. For example, a runway cannot be operated in the presence of strong crosswinds (relative to the orientation of that runway); or, a runway may not be adequately instrumented for operations under poor visibility conditions. There are also physical limits to the capacity of each configuration, i.e., to the number of arrivals and departures which may be accommodated in a given length of time while operating in a given configuration. In an operational context, capacity is typically measured as the expected number of movements that can take place in the presence of continuous demand. This is also known as the maximum throughput capacity and is measured as the number of arrivals and departures per unit of time, typically 10, 15, or 60 minutes. In making their decisions concerning the best runway configuration to use at any given time, controllers take into account the capacity of each available configuration, as well as the scheduled demand for arrivals and departures, and the weather forecast, which influences the future availability of the different runway configurations.

In our models, we represent the capacity of different configurations through the runway configuration capacity envelope (RCCE), studied and used in [22], [23], [27], and [24], and discussed in depth in [16]. An RCCE is a concave piecewise linear function which defines the set of feasible operating points that can be achieved under a given configuration and under given weather conditions. A runway configuration will typically have more than one RCCE, each corresponding to different weather conditions. Figure 2-2 shows two typical RCCE that may describe the capacity for arrivals and departures of a runway configuration like that shown in Figure 2-1. Note that the horizontal and vertical axes show, respectively, the number of departures and arrivals that are demanded or performed during the selected unit of time, be that 10, 15 or 60 minutes. The outermost RCCE shows the capacity available under Visual Meteorological conditions (VMC), while the inner one corresponds to the same configuration under Instrument conditions (IMC), when visibility is limited. Any integral point on or within the available RCCE corresponds to a feasible operating point, and any point outside is infeasible. For example, Point 4 is feasible in VMC, but infeasible in IMC. In the latter case, the configuration does not have sufficient capacity

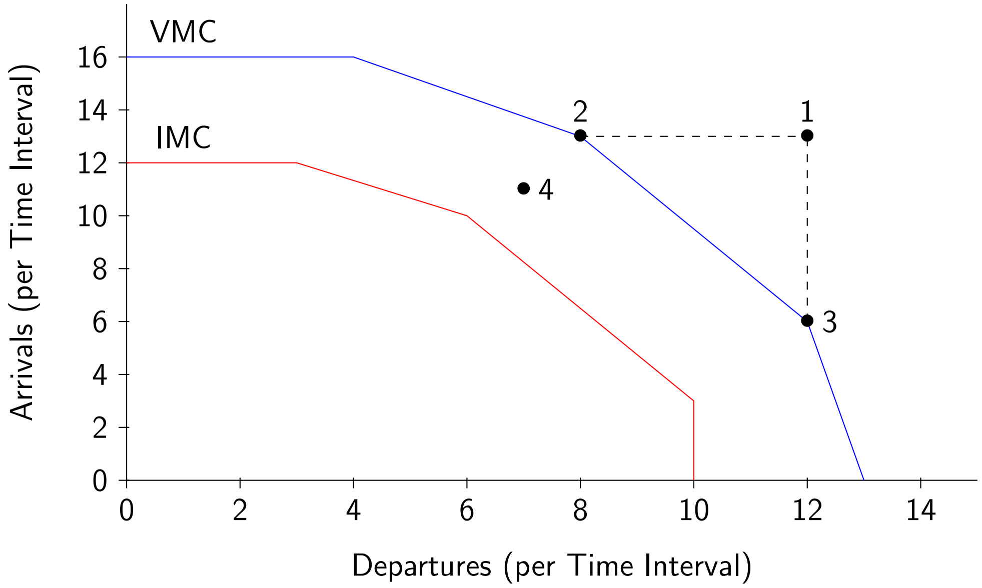

Figure 2-2: Runway configuration capacity envelopes (RCCE) for a single configuration under VMC and IMC.

to accommodate simultaneously the number of arrivals and departures associated with Point 4 during a single unit of time, resulting in the queueing of aircraft. The total number of RCCE that need to be considered, assuming the weather conditions to be known, is equal to the number of configurations, e.g., about 20 in the case of BOS.

In this chapter, we present mixed integer programming models (MIP), in which two naturally coupled problems are solved simultaneously. The first, the runway configuration management (RCM) problem, is to find a schedule of runway configurations which will best serve the demand over a specified time horizon. This is the problem which we have outlined thus far. Then, given this schedule of configurations, there remains a degree of freedom as to the exact balance of arrivals versus departures to serve during any time interval. Determining the optimal arrival/departure runway balancing (ADRB) is the second problem which controllers face. As mentioned in the Introduction, [23] and [24] proposed integer programs for the ADRB to minimize queueing by controlling the utilization of near-runway airspace and coordinating operations at arrival and departure fixes subject to a given schedule of configuration use. As far as we are aware, we present here the first optimization approach, not only to solve the RCM problem, but also the combined RCM and ADRB problems.

In practice, these problems are solved today largely on the basis of experience. Under this essentially "manual" approach, the FAA examines the weather forecast and the expected demand for the next several hours at each major airport and determines the sequence of runway configurations to be used, including the assignment of arrivals and departures to runways for each selected configuration. In doing so, location-specific issues, such as constraints on runway use imposed by noise or other considerations (see Section 2.5) are also taken into account. The difficult and most important instances occur when the weather is highly variable or poor (e.g., changing wind directions, occurrence of IMC). In such cases, the combination of changes in runway configurations – with attendant partial or full loss of capacity during changeover times – and of the need to select among "low" RCCE is likely to result in a significant backlog of arrivals and/or departures from one time period to the next. Under such conditions, the situation becomes highly dynamic and sequences and assignments may be modified frequently in response. The models described here can be most helpful in precisely these cases.

The solution to the RCM and ADRB problems may also impact air traffic flow management (ATFM) actions that the FAA and airlines take in the context of the Collaborative Decision Making (CDM) program. The solution essentially determines the "airport acceptance rate" (AAR) that ATFM uses to specify the number of arrivals that can be scheduled at each airport in each time period, i.e., the number of arrival "slots" that will be allocated under CDM among the airlines using the airport. If the AAR at an airport is low compared to the forecast demand for arrivals there, a ground delay program (GDP) may be initiated by ATFM for that airport, thus delaying before take-off (at their originating airport) flights headed to that airport. In this sense, the RCM and ADRB problems have implications not only at the local level, but also at a national one.

In solving these problems, our objective is to minimize the total weighted cost of delays to queued aircraft, where the weights might take into account the cost of fuel burn on the airport's surface (for departures) and in the near-airport airspace (for arrivals), as well as other considerations (e.g., controller workload) that may result in arrival delays being assigned a higher weight than delays to departures. Note that (i) the changeover times needed to transition between successive runway configurations and (ii) the restricted availability of certain configurations at certain times due to weather conditions are both critical aspects of the optimization to be performed. The former precludes the optimality of a greedy policy which simply operates in the best configuration for the demand in the current time interval (without taking into account the future sequence of configurations), while the latter requires that we consider the "lifespan" of each candidate configuration.

In our models, selecting a configuration is equivalent to selecting the outermost available RCCE corresponding to that configuration, this RCCE being determined by current weather conditions. (We assume that, for a given configuration, the IMC RCCE is always fully contained within the one for VMC.) Selecting the arrival/departure balance level is then equivalent to selecting an integral point on or within this RCCE. This selection also affects how delay, if any, is allocated between arrivals and departures. For instance, in Figure 2-2, let Point 1 correspond to the demand (d_1, a_1) during hour 1 and assume that the airport operates in VMC. Since Point 1 lies outside the available RCCE, some of the demand for the hour in question cannot be accommodated. In this particular instance, an optimal choice of operating point, considering only this hour, clearly lies somewhere on the line connecting Points 2 and 3.

We assume that the break points of each RCCE are integral, so that, along with nonnegativity constraints, the RCCE defines the convex hull of its feasible operating points in the two dimensional departures-arrivals operating space. These RCCE may be obtained by observing past operating points and fitting minimal concave piecewise linear functions which bound these points (see [22] and [28]). Alternatively, computational models of airport operations may be used to generate them, as discussed in [43], [16] and [42].

Contributions and Outline of Chapter

The most important contribution of this chapter is the development of mixed integer programming models to solve the RCM and ADRB problems efficiently for a single airport, as well as the proposed extension of the models to the metroplex case. In particular:

In Section 2.2, we present a MIP for the single airport problem, assuming that the changeover time between configurations is a constant, C, for all pairs of configurations. While this MIP does not define the convex hull of feasible solutions, several computational experiments on large realistic data sets in Section 2.2.3 indicate that the formulation is a strong one. Furthermore, the results of these experiments indicate that the potential benefit of using this approach is significant, and that improvements of the order of at least 10% could be expected relative to a sophisticated optimization-based heuristic.

In Section 2.3, we relax the above assumption to allow the changeover time between any two configurations to belong to a set of the form $\{iC : i = 1, \ldots, \nu\}$, where C is a constant. Computational experiments show that for $\nu \in \{2, 3\}$, optimal solutions are still obtained within several minutes.

In Section 2.4, we present an extension of the model to optimize over a metroplex of airports.

In Section 2.5, we show that important environmental constraints can be easily incorporated into the model.

2.2 A Mixed Integer Programming Model for a Single Airport with Constant Configuration Changeover Times

Here we seek to solve the RCM and ADRB problems, outlined in the Introduction, for a single airport. We assume that the time required to change between configurations is a constant, C, and that the relevant data, as listed in the definitions below, are available and known with certainty. The latter assumption is reasonable given that we consider in our model a short time horizon of a few hours. Furthermore, we envisage the method being implemented using a rolling horizon approach, re-solving every 30 minutes or so in order to account for any divergence of observed from expected data.

$\mathcal{T}$ = the set of time intervals, $\{1, 2, \ldots, T\}$;

$\mathcal{K}_t$ = the set of configurations available during time interval t;

$\mathcal{J}_k^t$ = the set of linear pieces of the outermost RCCE available for configuration k at time t;

a_t = the number of arrivals scheduled for time interval t;

d_t = the number of departures scheduled for time interval t;

c_t = the cost of delaying a single arrival for one time interval at time interval t;

q_t = the cost of delaying a single departure for one time interval at time interval t.

In addition, we let the j^{th} piece of the outermost RCCE available for configuration k at time t, an affine function, be defined by the parameters $\alpha_{jk} > 0$, $\beta_{jk} \leq 0$ and $\gamma_{jk} \geq 0$. While these parameters depend on t, since j belongs to a set which depends on t, we choose to omit this from our notation.

Note that the time horizon is discretized such that the length of one time interval is equal to the changeover time, C, which is typically of the order of 10 minutes. In this way, we can model a changeover by allowing no arrivals or departures to be served during the time interval at which the changeover occurs. Moreover, the set of time intervals, $\mathcal{T}$, includes a number of time intervals at the end of the original time horizon, with no scheduled arrivals or departures. During these added time intervals any queues which have built up can be cleared. We assume that we are able to clear all queues without having to cancel or re-route any flights.

Our decision variables are the following, and are defined for every t only for those configurations k which are available:

$$z_{kt} = \begin{cases} 1, & \text{if we operate in configuration } k \text{ at time } t, \\ 0, & \text{otherwise;} \end{cases}$$

$$y_{kt} = \begin{cases} \text{the number of arrivals served at time } t, & \text{if in configuration } k \text{ at } t, \\ 0, & \text{otherwise;} \end{cases}$$

$$x_{kt} = \begin{cases} \text{the number of departures served at time } t, & \text{if in configuration } k \text{ at } t, \\ 0, & \text{otherwise;} \end{cases}$$

$u_t =$ the number of arrivals which go unserved at time t;

$v_t =$ the number of departures which go unserved at time t.

The mixed integer optimization problem is as follows:

$$\begin{aligned}
\textbf{P2-1:} \quad \min \quad & \sum_{t \in \mathcal{T}} (c_t u_t + q_t v_t) \\
\text{s.t.} \quad & u_t - u_{t-1} + \sum_{k \in \mathcal{K}_t} y_{kt} = a_t, \quad \forall t \in \mathcal{T}, && (2.1a) \\
& v_t - v_{t-1} + \sum_{k \in \mathcal{K}_t} x_{kt} = d_t, \quad \forall t \in \mathcal{T}, && (2.1b) \\
& \gamma_{jk} y_{kt} - \beta_{jk} x_{kt} - \alpha_{jk} z_{kt} \leq 0, \quad \forall j \in \mathcal{J}_k^t, \forall k \in \mathcal{K}_t, \forall t \in \mathcal{T}, && (2.1c) \\
& \sum_{k \in \mathcal{K}_t} z_{kt} \leq 1, \quad \forall t \in \mathcal{T}, && (2.1d) \\
& \sum_{k' \in \mathcal{K}_{t-1} \setminus \{k\}} z_{k',t-1} + z_{kt} \leq 1, \quad \forall k \in \mathcal{K}_t, \forall t \in \mathcal{T} \setminus \{1\}, && (2.1e) \\
& z_{kt} \in \{0,1\}, \quad \forall k \in \mathcal{K}_t, \forall t \in \mathcal{T}, && (2.1f) \\
& u_t, v_t \in \mathbb{Z}_+, \quad \forall t \in \mathcal{T}, && (2.1g) \\
& y_{kt} \ \geq \ 0, \quad \forall k \in \mathcal{K}_t, \forall t \in \mathcal{T}, && (2.1h) \\
& x_{kt} \ \geq \ 0, \quad \forall k \in \mathcal{K}_t, \forall t \in \mathcal{T}, && (2.1i) \\
& u_0, v_0 = 0. && (2.1j)
\end{aligned}$$

Our objective is to minimize the total cost of delays incurred. Constraints (2.1a), (2.1b), (2.1g) and (2.1j) define the variables u_t and v_t, the number of arrivals and departures, respectively, which go unserved at time t. Constraints (2.1c) force the operating point, given that we operate in the k^{th} configuration at time t, to lie within its RCCE, while forcing it to zero if $z_{kt} = 0$. Constraints (2.1d) state that at any time t, we may operate in at most one configuration.

Constraints (2.1e) invoke our fundamental assumption that the changeover time is equal to the length of one time interval. In this way, we model the cost of changing from one configuration to another by enforcing a delay of one time interval. In other words, it is not possible to operate in configuration k at time t and also in configuration $k' \neq k$ at time $t-1$.

One could also modify Constraints (2.1e) to consider only a subset of all pairs of configurations $\{k, k'\}$, thus modeling the changeovers between the excluded pairs as being instantaneous. This would be suitable if the corresponding changeover times were very small compared to other changeovers.

Note that we have defined the variables u_t and v_t to be integral in Constraints (2.1g). Given our assumptions on the nature of the RCCE and the integrality of the arrival and demand data, as well as Constraints (2.1f) on z_{kt}, it follows that in an optimal solution to the resulting MIP, y_{kt} and x_{kt} are integral. We therefore relax the integrality constraints for these two sets of variables and greatly reduce the number of integer variables in the model.

We can then add two classes of valid inequalities in order to strengthen the formulation. First, we add Inequalities (2.2) below, which are closely related to Constraints (2.1e). Observe that in Inequalities (2.2), we require for all k that we cannot both operate in configuration k at time $t-1$ and also in some configuration $k' \neq k$ at time t, while in Constraints (2.1e), we require for all k that we cannot both operate some configuration $k' \neq k$ at time $t-1$ and in configuration k at time t.

$$z_{k,t-1} + \sum_{k' \in \mathcal{K}_t \setminus \{k\}} z_{k't} \leq 1, \quad \forall k \in \mathcal{K}_{t-1}, \forall t \in \mathcal{T} \setminus \{1\}. \tag{2.2}$$

It is clear that, while the set of feasible integer solutions remains unchanged under this addition, we remove some non-integral solutions from the solution set of the LP relaxation of P2-1.

Now we generate for each time interval t a single RCCE which defines the convex hull of the set $\{(x, y) \in \mathbb{Z}_+^2 : \exists k \in \mathcal{K}_t \text{ s.t. } \gamma_{jk} y - \beta_{jk} x \leq \alpha_{jk},\ \forall j \in \mathcal{J}_k^t\}$, i.e. the minimal piecewise linear concave envelope which majorizes all RCCE at time t. We let this RCCE be defined by the parameters α'_{jt}, β'_{jt} and γ'_{jt}, $\forall j \in \mathcal{J}'_t$. Then, observing that $y_t \triangleq \sum_{k \in \mathcal{K}_t} y_{kt}$ and $x_t \triangleq \sum_{k \in \mathcal{K}_t} x_{kt}$ represent the number of arrivals and departures served at time t, respectively, it is clear that (x_t, y_t) must lie within this RCCE, and hence Inequalities (2.3) below are valid.

$$\gamma'_{jt} \sum_{k \in \mathcal{K}_t} y_{kt} - \beta'_{jt} \sum_{k' \in \mathcal{K}_t} x_{k't} \leq \alpha'_{jt}, \quad \forall j \in \mathcal{J}'_t, \forall t \in \mathcal{T}. \tag{2.3}$$

Furthermore, these inequalities give the tightest possible bound on the relation between feasible y_t and x_t, since the convex hull of this (in general non-convex) set has been defined.

2.2.1 Example Problem

We now present a simple example in order to test the model and gain insight into its solution. We consider a time horizon of 10 periods of demand, set $T = 15$ to allow time to serve all demand, and $c_t = 12$, $q_t = 10$, $\forall t \in \mathcal{T}$ in order to capture the typically greater cost of delaying arrivals compared to departures. In all examples that follow, we shall use these cost coefficients. The scheduled demand for arrivals and departures is displayed in Figure 2-3 along with the RCCE corresponding to three configurations, which are available throughout the entire time horizon.

This problem, along with all others that follow in this chapter, was solved using AMPL CPLEX 11.2.10, using a single thread, on a computer with an Intel(R) Core(TM) 2 Duo E7400 Processor (2.80GHz, 3MB Cache, 1066MHz FSB) and 2GB of RAM, running Ubuntu Linux. The optimal solution, which is displayed in more detail in Figures 2-4, 2-5 and 2-6, was found in 0.04 seconds, and consists of operating in configuration 2 for intervals 1 and 2, configuration 3 for intervals 4 to 6, and then

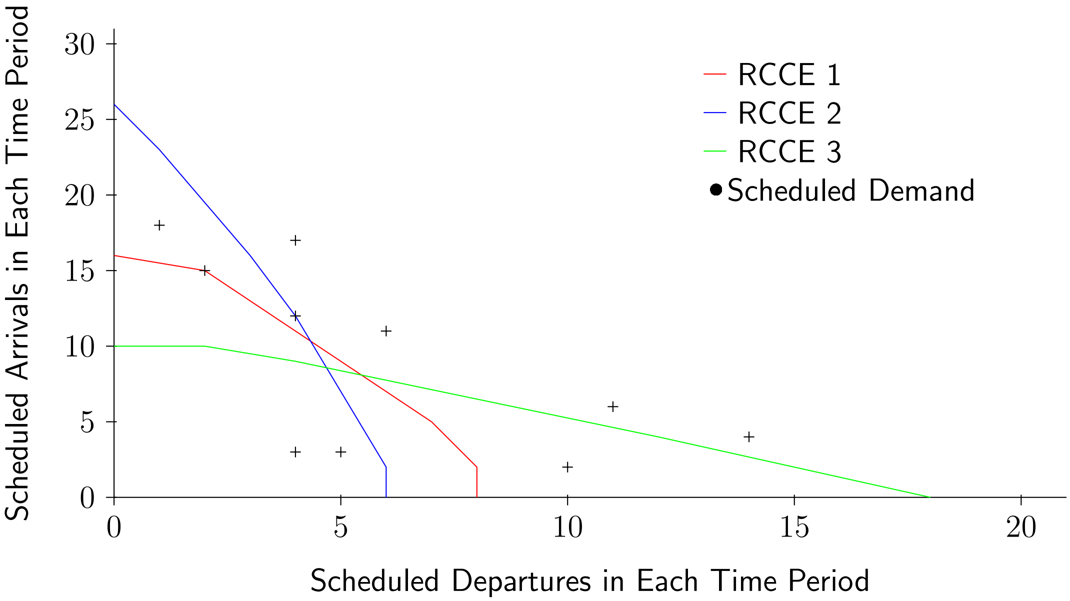

Figure 2-3: Test Problem 1 data, with 3 configurations and 15 time intervals. The scheduled demand is indicated for intervals 1 to 10, and is zero for intervals 11 to 15.

in configuration 2 again from interval 8 onwards.

Note that the "actual demand" in the system at time t, given the operating policy in the preceding time intervals, consists of $u_{t-1} + a_t$ arrivals and $v_{t-1} + d_t$ departures, i.e., any arrivals (departures) left in queue at the end of time interval $t-1$ plus the scheduled arrivals (departures) in time interval t. We can then observe the following:

1. In Figure 2-4, the actual demand and scheduled demand are identical, since the first demand point lies within RCCE 2, with which we operate, and hence all demand is served and there is no backlog added to the second time interval.

2. In Figure 2-5, since we are changing from configuration 2 to configuration 3, we have $x_3 = y_3 = 0$. In this way, the third actual demand point is added to the fourth scheduled demand point to create the fourth actual demand point. Similar behavior is also seen with time intervals 7 and 8 in Figure 2-6.

3. When a change is made, it is made during an interval of relatively low actual demand and before a sequence which is favored by the new configuration.

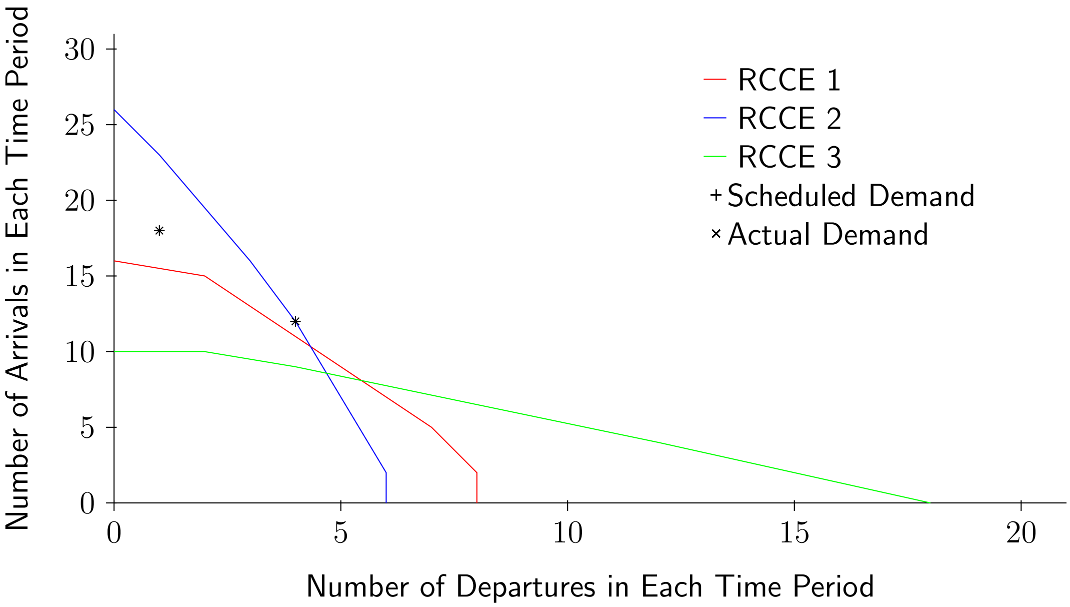

Figure 2-4: Test Problem 1 solution, intervals 1 and 2.

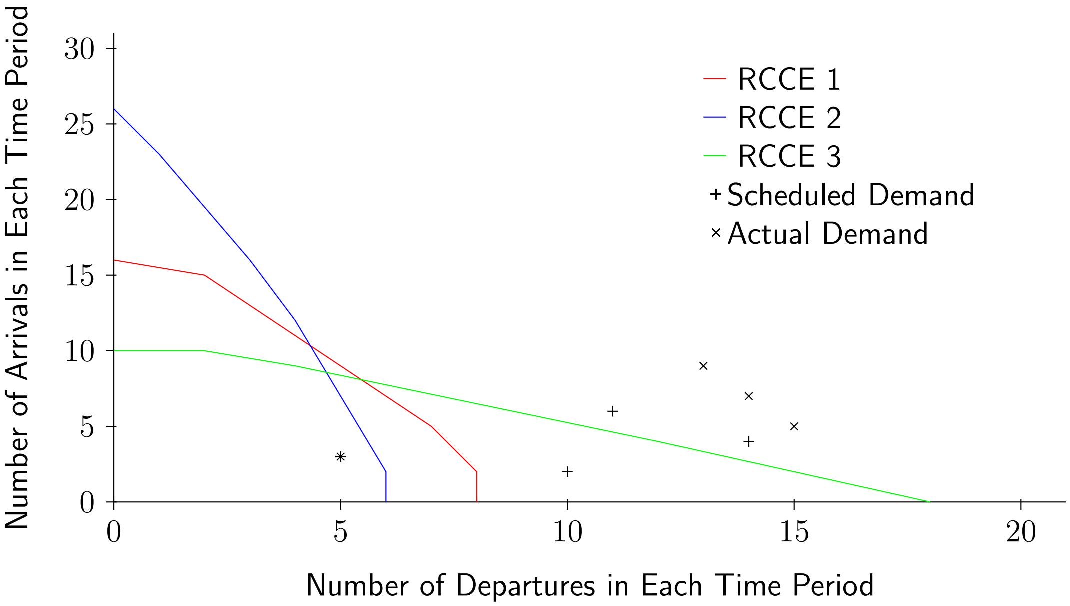

Figure 2-5: Test Problem 1 solution, intervals 3 to 6.

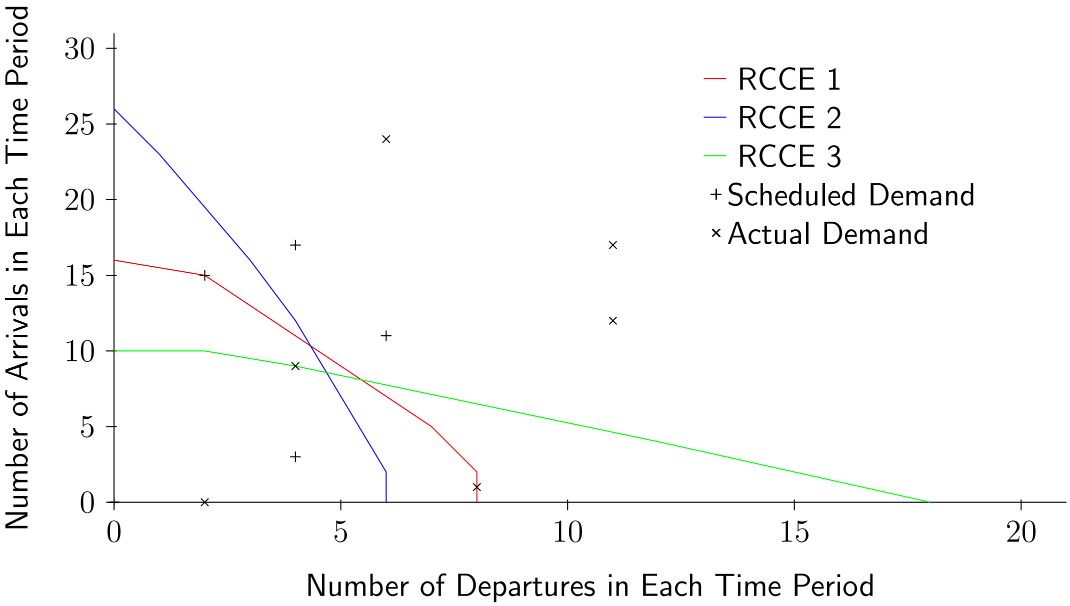

Figure 2-6: Test Problem 1 solution, intervals 7 to 12.

4. As time passes, the scheduled and actual demand may diverge significantly.

In summary, we have learned that, even for a very small problem, it is essential to account for the cumulative nature of demand and its dependence on our decisions; our early decisions may have a long-lasting impact on the overall problem. These interactions may be very complex, and hence good decision-making will in general be difficult without the use of sophisticated tools.

2.2.2 A Baseline Policy

Before performing computational experiments on problems of realistic size, we now devote our attention to presenting a baseline policy in order to obtain a reasonable indication of the relative quality of the solutions obtained from P2-1 compared with, say, current practice. In related literature, [37] developed a maximum-likelihood discrete-choice model to describe the configuration change process. We here present a baseline policy designed (i) to obtain an estimate of a lower bound on the improvement that could be observed in practice by implementing the policy obtained from solving P2-1 and (ii) to mimic the approach that may be taken by a highly skilled controller.

The method is developed through a “smart” heuristic summarized below, first at a high level, and then in detail.

1. Among all configurations which are available for a significant period of time starting now, choose the one that is best (per the criteria described below) to operate in.

2. Operate in this configuration until the first time period at which it is no longer available for use.

3. When that happens, observe the new state of the system: if there is no demand left, then stop; otherwise, return to Step 1.

The algorithm is essentially a greedy algorithm, modified to avoid a large number of costly changeovers. We believe this to be a good approximation of controllers' actions: given the difficulty of foreseeing the effect of one's decisions on future demand, one cannot plan “manually” a good configuration sequence too far into the future.

In order to choose which configuration is selected as “best,” we first restrict our list of eligible configurations to those which are available for a reasonably long period of time, preventing too many forced changeovers. We then solve a relevant optimization problem, P2-2(t), shown below, over these configurations. P2-2(t) is an advanced optimization problem and, as a result, the RCM and ADRB policies that are developed by the baseline heuristic are sophisticated. We are confident that, on average, any improvement in objective obtained through P2-1 over a policy resulting from the baseline heuristic will represent a lower bound on the improvement that would be observed in practice by implementing a policy obtained from P2-1.

In making such a claim, one cannot ignore the stochastic environment in which this problem is solved. Recall, however, that we consider a short time horizon of a few hours, limiting the uncertainty associated with the data. Moreover, implementation of our methodology should use a rolling horizon approach, re-solving the MIP every 15 minutes or so, in order to take into account changing conditions, while always looking far enough ahead into the future to avoid simply relying on a greedy policy.

The following algorithm computes and simulates the baseline policy:

Algorithm 2.1. 1. Set $t := 1$, and let $f := 0$ be the simulated cost, $\bar{a}_\tau := a_\tau$ the actual demand for arrivals and $\bar{d}_\tau := d_\tau$ the actual demand for departures, $\forall \tau \in \mathcal{T}$. Also let η be the default number of consecutive time periods for which a configuration must be available in order to be considered for selection. Here, we let $\eta = 6$, corresponding to one hour.

2. If $\mathcal{K}_t \neq \emptyset$, then go to Step 3; else, no demand is served at this time period, so update the simulated cost and actual demand by setting $f := f + c_t\bar{a}_t + q_t\bar{d}_t$, $\bar{a}_{t+1} := \bar{a}_{t+1} + \bar{a}_t$, $\bar{d}_{t+1} := \bar{d}_{t+1} + \bar{d}_t$ and $t := t + 1$. If $t \leq T$, then go to Step 2; else, Stop.

3. Let $\bar{\mathcal{K}}_t = \cap_{h=0}^{\eta-1}\mathcal{K}_{t+h}$ be the set of configurations eligible for selection. If $\bar{\mathcal{K}}_t = \emptyset$, then set $\eta := \eta - 1$ and go to Step 3; else, set $\eta := 6$ and go to Step 4.

4. Given the state of the system (note that the current scheduled demand will have been updated to include any currently enqueued traffic), choose the configuration to operate in by solving the MIP P2-2(t) if $t = 1$, or P2-2($t + 1$) otherwise:

$$\mathbf{P2\text{-}2(t)}: \quad \min \quad \sum_{\tau=t}^{T}(c_\tau u_\tau + q_\tau v_\tau)$$

$$\text{s.t.} \quad u_\tau + \sum_{k\in\bar{\mathcal{K}}_t} y_{k\tau} = a_\tau, \quad \forall \tau \in \{t, \ldots, T\}, \tag{2.4a}$$

$$v_\tau + \sum_{k\in\bar{\mathcal{K}}_t} x_{k\tau} = d_\tau, \quad \forall \tau \in \{t, \ldots, T\}, \tag{2.4b}$$

$$\gamma_{jk}y_{k\tau} - \beta_{jk}x_{k\tau} - \alpha_{jk}z_{k\tau} \leq 0,$$
$$\forall j \in \mathcal{J}_k^t, \forall k \in \bar{\mathcal{K}}_t, \forall \tau \in \{t, \ldots, T\}, \tag{2.4c}$$

$$z_{k\tau} - \theta_k \leq 0, \quad \forall k \in \bar{\mathcal{K}}_t, \forall \tau \in \{t, \ldots, T\}, \tag{2.4d}$$

$$\sum_{k\in\bar{\mathcal{K}}_t} \theta_k \leq 1, \tag{2.4e}$$

$$z_{k\tau} \in \{0, 1\}, \quad \forall k \in \bar{\mathcal{K}}_t, \forall \tau \in \{t, \ldots, T\}, \tag{2.4f}$$

$$u_\tau, v_\tau \in \mathbb{Z}_+, \quad \forall \tau \in \{t, \ldots, T\}, \tag{2.4g}$$

$$y_{k\tau} \;\geq\; 0, \quad \forall k \in \bar{\mathcal{K}}_t, \forall \tau \in \{t, \ldots, T\}, \qquad (2.4h)$$

$$x_{k\tau} \;\geq\; 0, \quad \forall k \in \bar{\mathcal{K}}_t, \forall \tau \in \{t, \ldots, T\}. \qquad (2.4i)$$

Note that Constraints (2.4d) and (2.4e) mean that only one configuration can be chosen for the entire time horizon $t, \ldots, T$. Furthermore, only current scheduled demand is considered for any given time period, i.e., the backlogging of demand in future time periods is not taken into account in Constraints (2.4a) and (2.4b), unlike in the related Constraints (2.1a) and (2.1b) of P2-1. The underlying rationale is that the effect of controllers' decisions on future demand may be very hard to predict (as shown in the example problem of Section 2.2.1), and so this should more closely resemble the real-world decision-making approach of controllers.

Hence P2-2(t) chooses the *single* best configuration according to *scheduled* demand over the remaining time horizon. We let k^* be this configuration, i.e. such that $\theta^*_{k^*} = 1$, where θ^* is the optimal vector θ of P2-2(t), and let $t' = \min\{\tau : \tau > t, k^* \notin \mathcal{K}_\tau\}$ be the first time at which configuration k^* becomes unavailable (note that this will be $T + 1$ if k^* is available for the rest of the time horizon). We will operate in configuration k^* until time $t' - 1$.

5. If $t \neq 1$, then let $\delta := 0$, preventing demand from being served at time t through Inequalities (2.5c) in P2-3(k^*, δ, t, t') below; else, let $\delta := 1$.

6. Now simulate the effect of this choice of k^* for the interval $[t, t' - 1]$. Solve the following MIP, P2-3(k^*, δ, t, t'):

$$\mathbf{P2\text{-}3(k^*, \delta, t, t')} : \quad \min \quad \sum_{\tau=t}^{t'-1} (c_\tau u_\tau + q_\tau v_\tau)$$

$$\text{s.t.} \quad u_\tau - u_{\tau-1} + y_\tau = \bar{a}_\tau, \quad \forall \tau \in \{t, \ldots, t' - 1\}, \qquad (2.5a)$$

$$v_\tau - v_{\tau-1} + x_\tau = \bar{d}_\tau, \quad \forall \tau \in \{t, \ldots, t' - 1\}, \qquad (2.5b)$$

$$\gamma_{jk^*} y_t - \beta_{jk^*} x_t - \delta.\alpha_{jk^*} z_t \leq 0, \quad \forall j \in \mathcal{J}^t_{k^*}, \qquad (2.5c)$$

$$\gamma_{jk^*} y_{k^*\tau} - \beta_{jk^*} x_\tau - \alpha_{jk^*} z_\tau \leq 0,$$

$$\forall j \in \mathcal{J}_{k^*}^{\tau}, \forall \tau \in \{t+1, \ldots, t'-1\}, \quad (2.5d)$$

$$z_\tau \in \{0,1\}, \quad \forall \tau \in \{t, \ldots, t'-1\}, \quad (2.5e)$$

$$u_\tau, v_\tau \in \mathbb{Z}_+, \quad \forall \tau \in \{t, \ldots, t'-1\}, \quad (2.5f)$$

$$y_\tau \ \geq \ 0, \quad \forall \tau \in \{t, \ldots, t'-1\}, \quad (2.5g)$$

$$x_\tau \ \geq \ 0, \quad \forall \tau \in \{t, \ldots, t'-1\}, \quad (2.5h)$$

$$u_{t-1}, v_{t-1} = 0. \quad (2.5i)$$

The above MIP simply fixes the configuration which is operated over the time interval $[t, t'-1]$ to be k^*, and simulates the effect of this decision, forcing zero demand to be served at time t if $\delta = 0$.

7. Update the simulated cost and actual demand by setting $f := f + \sum_{\tau=t}^{t'-1}(c_\tau u_\tau^* + q_\tau v_\tau^*)$, $\bar{a}_{t'} := \bar{a}_{t'} + u_{t'-1}^*$, and $\bar{d}_{t'} := \bar{d}_{t'} + v_{t'-1}^*$.
8. Set $t := t'$. If $t \leq T$, then go to Step 2.
9. Stop. The baseline value is f.

2.2.3 Computational Results for Larger Problems

Now we introduce two problems of realistic size, which are shown in Figures 2-7 and 2-8. We then report corresponding computational experience, including on a third problem for which we don't present a figure, being too complex to be worth trying to visualize. In these computational experiments, we seek to understand i) whether the model is computationally tractable, and ii) how solutions obtained from the model might compare with current practice, as approximated by our baseline.

We note that the set of RCCE in Problem 4 is a superset of that in Problem 3, which is itself a superset of that in Problem 2. Each problem has 30 time intervals, representing 24 10-minute intervals, and six intervals at the end with zero demand. Table 2.1 outlines the availability of configurations. In addition, when a configuration is available, some of its RCCE may be unavailable, but these details are omitted for brevity.

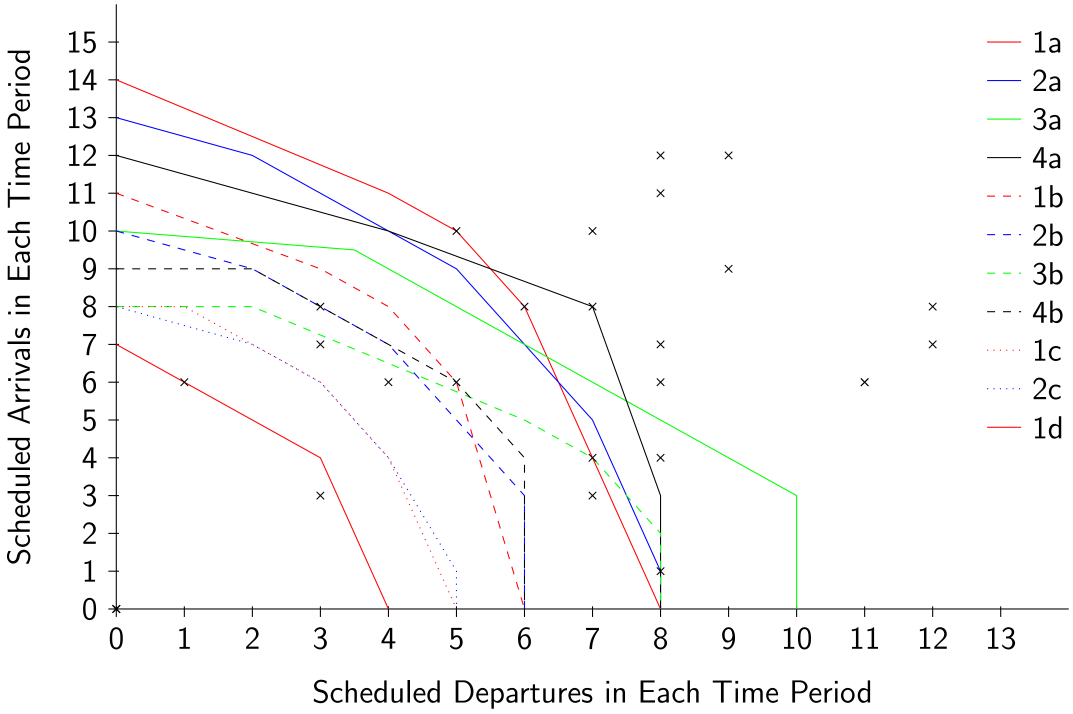

Figure 2-7: Test Problem 2, with 4 configurations, 11 RCCE and 30 time intervals. In the legend "2c" indicates the third RCCE for configuration 2, etc.

Problem	**Configurations which are Unavailable**
2	1 at $t = 20$; 2 on $[19, 21]$; 3 on $\{6, 12, 25\}$; 4 on $\{5, 6, 12, 13, 25\}$
3	1 and 5 on $[16, 17]$; 2 and 6 on $[15, 17]$; 3 and 7 on $\{4, 27\}$; 4 and 8 on $\{4, 5, 27\}$
4	1, 5, 9 and 11 on $[16, 17]$; 2 and 6 on $[15, 17]$; 3 and 7 on $\{4, 27\}$; 4, 8 and 12 on $\{4, 5, 27\}$; 10 on $\{4, 15, 27\}$

Table 2.1: Availability of configurations for problems 2–4.

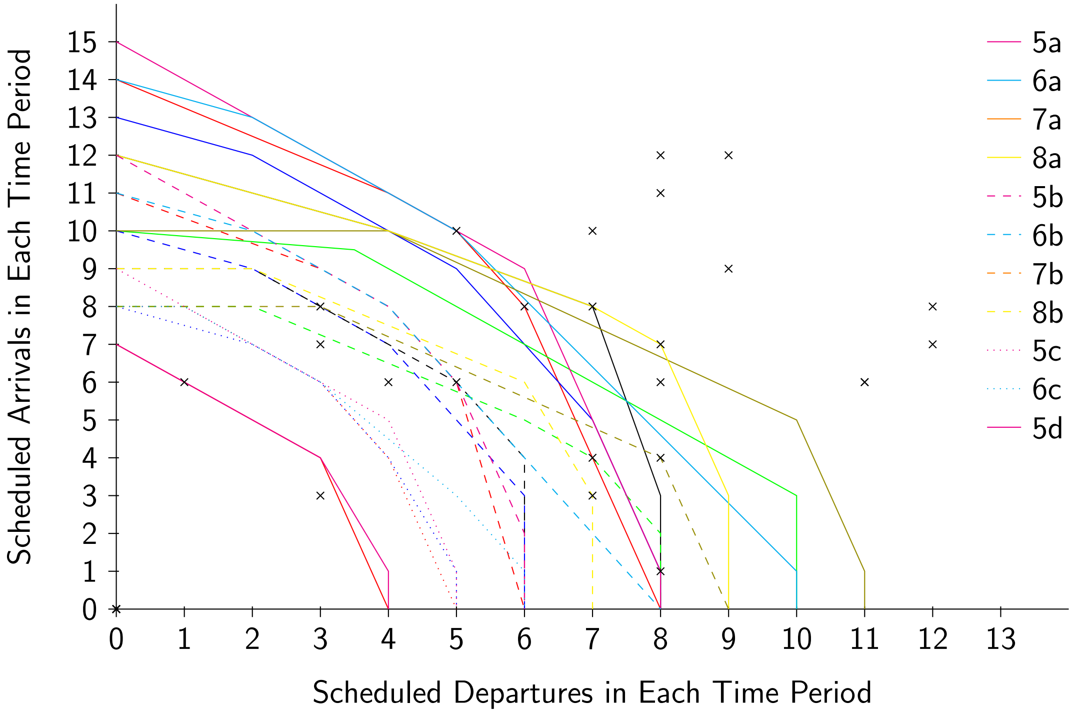

Figure 2-8: Test Problem 3, with 8 configurations, 22 RCCE and 30 time intervals. The colors and styles for Configurations 1 – 4 and their RCCE are the same as in Figure 2-7.

Problem	**Computation Time (s)**	$\mathbf{Z^*_{IP}}$	$\mathbf{Z^*_{LP}}$	**% Difference between** $\mathbf{Z^*_{IP}}$ **and** $\mathbf{Z^*_{LP}}$
2	0.1	6148	5681.4	7.6
3	1.1	4752	3376.4	28.9
4	2.6	3330	2788.7	16.2

Table 2.2: Computation times for RCM-ADRB and a measure of the strength of the formulation.

Problem	**Configurations which are Unavailable**
5	1 at $10, 17$ and 18; 2 at 21; 5 at $11, 20$ and 21; 6 on $[10, 12]$, at 20 and 21; 9 at 21
6	$3, 4, 5, 6, 7$ and 8 on $[10, 11]$; $1, 2, 3, 4, 5$ and 6 on $[20, 21]$
7	$2, 3$ and 4 on $[13, 14]$; $6, 7$ and 8 at 25

Table 2.3: Availability of configurations for problems 5–7.

Table 2.2 shows that computation times were in all cases less than 3 seconds, even for the largest of problems. Indeed, another indication of the strength of the model, also shown in the table, is the small difference between the optimal objective value of RCM-ADRB, Z^*_{IP}, and the optimal value of the LP relaxation of RCM-ADRB, Z^*_{LP}.

We now compare the policies and objective values obtained from RCM-ADRB to those of the baseline heuristic. The results are presented in Tables 2.4 and 2.5. Note that we have introduced further examples $5 - 7$, with 9, 8 and 9 configurations, respectively, whose detailed description we omit for brevity. We do, however, describe the availability of these RCCE in Table 2.3.

It can be observed that in many cases our model far outperforms the baseline approach, by margins of $10 - 50\%$. This can be explained by the cumulative nature of the demand in the system, which amplifies the negative effect of any sub-optimal decisions made early in the time horizon. However, we also note that in one instance, where the policies are similar, the improvement is only 2%.

A small difference could also be expected when all configurations are available at all time periods, since in this case the baseline heuristic would produce a policy which uses a single configuration, and P2-1 would also be likely to do the same,

Problem	**RCM-ADRB Solution Value**	**Baseline Solution Value**	**% Difference**
2	6148	6964	13.3
3	4752	7180	51.1
4	3330	3396	2.0
5	8462	12510	47.8
6	8826	10504	19.0
7	10078	11638	15.5

Table 2.4: Comparison of solutions from the model RCM-ADRB with baseline solutions.

despite its slight difference from P2-2(1). However, in this case the heuristic almost becomes a complete optimization. Furthermore, this case is an uninteresting one, since in practice many configurations can often not be operated for prolonged periods of time. This problem becomes most interesting when some configurations become unavailable and we must determine whether to choose the one which is best now, even though it will become unavailable at a later point, or to choose some other configuration which will remain available for a longer period of time. In any case, it is clear that the potential benefit from the implementation of this method may be significant. Overall, we expect to see an average improvement of the order of at least 10% in practice, given the results presented here, and the advanced nature of our baseline heuristic, which involves solving a relevant optimization problem.

2.3 Multiple Changeover Times

We now extend the model to allow different changeover times belonging to a finite set of the form $\{iC : i = 1, \ldots, \nu\}$. We must first discretize the time horizon further, such that the length of one time interval is equal to the minimum changeover time, C. For a given problem solved by P2-1, this will increase the number of time intervals in the time horizon and scale down the RCCE, both by a factor of ν. Note that the new RCCE will then not necessarily define the convex hull of the feasible operating points, hence all such RCCE must be rounded so that they do indeed satisfy this requirement.

Problem	**RCM-ADRB Policy**	**Baseline Policy**
2	1 on $[1, 17]$; 4 on $[19, 24]$; 3 on $[26, 30]$	1 on $[1, 19]$; 1 on $[21, 30]$
3	1 on $[1, 5]$; 4 on $[7, 26]$;	6 on $[1, 14]$; 4 on $[16, 26]$; 4 on $[28, 30]$
4	11 on $[1, 15]$; 10 on $[17, 26]$	11 on $[1, 15]$; 4 on $[17, 26]$
5	9 on $[1, 19]$; 1 on $[21, 30]$	1 on $[1, 9]$; 1 on $[11, 16]$; 4 on $[18, 30]$
6	2 on $[1, 19]$; 8 on $[21, 30]$	8 on $[1, 9]$; 2 on $[11, 19]$; 7 on $[21, 30]$
7	2 on $[1, 12]$; 6 on $[14, 23]$; 5 on $[25, 30]$	3 on $[1, 12]$; 8 on $[14, 24]$; 8 on $[26, 30]$

Table 2.5: Comparison of the optimal policies and the baseline policies.

However, we cannot make the discretization too fine, since the rounding error of these RCCE will then become significant.

In order to model changeover times, for cases in which one time interval now becomes two, we let $\mathcal{K}' = \{(k_1, k_2), (k_3, k_4), \ldots, (k_{m-1}, k_m)\}$ be the set of ordered pairs representing changeovers which take two time intervals, where $k_i \in \cup_{t\in\mathcal{T}}\mathcal{K}_t$, $\forall i \in \{1, \ldots, m\}$. We then add the following constraints, of which there are at most $m(T-2)/2$, to the model:

$$z_{k,t-2} + \sum_{k' \colon (k,k') \in \mathcal{K}'} z_{k't} \leq 1, \quad \forall k \text{ s.t. } \exists (k, k') \in \mathcal{K}', \forall t \in \mathcal{T} \setminus \{1, 2\}. \tag{2.6}$$

For three changeover times, which for all practical purposes is the most that would be necessary, we first triple the number of time intervals in the time horizon and then add the constraints for those changeovers which take two time intervals, exactly as in (2.6) above. In addition, we let $\mathcal{K}'' = \{(k_1, k_2), (k_3, k_4), \ldots, (k_{n-1}, k_n)\}$ be the set of ordered pairs representing changeovers which take three time intervals. We then add the following constraints to the model:

$$z_{k,t-3} + \sum_{k' \colon (k,k') \in \mathcal{K}''} z_{k't} \leq 1, \quad \forall k \text{ s.t. } \exists (k, k') \in \mathcal{K}'', \forall t \in \mathcal{T} \setminus \{1, 2, 3\}. \tag{2.7}$$

Change- over Times	Integer Variables	Continuous Variables	Upper Bound on Number of Constraints
1	$T(2+K)$ (420)	$2KT$ (720)	$T(3+J+JK+4K)$ (2700)
2	$2T(2+K)$ (840)	$4KT$ (1440)	$2T(3+J+JK+4K+2K^2)$ (22700)
3	$3T(2+K)$ (1260)	$6KT$ (2160)	$3T(3+J+JK+4K+4K^2)$ (60000)

Table 2.6: Effect of the number of changeover times on the size of the model. T is the length of the time horizon under a single changeover time. Figures in parentheses correspond approximately to the largest problem considered in Section 2.2.3, with $J = 3$, $K = 12$, $T = 30$. Valid inequalities are included in these counts.

Finally, we can again add valid inequalities in the spirit of Inequalities (2.2), corresponding to both Constraints (2.6) and (2.7).

2.3.1 Size of the Model

We now present in Table 2.6 the effect of allowing multiple changeover times on the size of the model. To simplify, we assume that $\forall k \in \mathcal{K}_t$, $\forall t \in \mathcal{T}$, we have $|\mathcal{K}_t| = K$ and $|\mathcal{J}_k^t| = |\mathcal{J}_t'| = J$.

Note that in the worst case, allowing for three changeover times increases the number of constraints by a factor of 20, but the model is still not excessively large. This case occurs when there is a single changeover which is about a third of the length of all others.

2.3.2 Computational Results

In the interest of estimating computation times for the case of multiple changeover times, as well as gaining some insight into the effect of this modeling adjustment on optimal solutions, we next modify the three problems from Section 2.2.3. We first consider the case of two changeover times, allowing the changeovers between two configurations to take half the time of all other changeovers. These shorter

Problem	Short Changeovers	Optimal Policy	Time (s)
2′	$\{1, 3\}$	1 on $[1, 35]$, 3 on $[37, 48]$, 1 on $[50, 56]$	0.5
3′	$\{2, 3\}$	2 on $[1, 7]$, 3 on $[9, 51]$	24.2
4′	$\{4, 11\}$	11 on $[1, 18]$, 4 on $[20, 50]$	4.9

Table 2.7: Policies and computation times for two changeover times.

changeovers are indicated in Table 2.7, along with the corresponding optimal policies and computation times.

From Table 2.7, it can be seen that computation times increased by an order of magnitude over those of Table 2.2 in Section 2.2.3, but were still very short. Furthermore, different optimal policies are obtained which favor the shorter changeovers, compared to those of Table 2.5 in Section 2.2.3. In particular, the optimal configuration sequence for Problem 2 changes from $(1, 4, 3)$ to $(1, 3, 1)$, while that for Problem 3 changes from $(1, 4)$ to $(2, 3)$, and that for Problem 4 from $(11, 10)$ to $(11, 4)$, each clearly taking advantage of the shorter changeover times, as might be expected.

Next we modify the problems from Section 2.2.3 in a similar way to allow for three different changeover times. The modifications and results are shown in Table 2.8. Again, the optimal policies change due to the different changeover times between certain configurations. For example, comparing with Table 2.7, the optimal configuration sequence for Problem 2 changes from $(1, 3, 1)$ to $(1, 3, 2)$, given the short changeover between configurations 2 and 3. Furthermore, the long changeover between configurations 1 and 4 rules out the optimal solution $(1, 4, 3)$ to the original problem, shown in Table 2.5, Section 2.2.3.

While it is clear that computation times are significantly longer than in the case of one or two changeover times, these results indicate that optimal solutions are obtained quickly enough to warrant successful implementation in practice, even for problems of the largest size considered here.

Problem	Short	Long	Optimal Policy	Time (s)
2′′	$\{2, 3\}$	$\{1, 4\}$	1 on $[1, 52]$, 3 on $[55, 72]$, 2 on $[74, 84]$	2.4
3′′	$\{3, 4\}$	$\{1, 4\}$	2 on $[1, 16]$, 4 on $[19, 77]$	122.1
4′′	$\{4, 10\}$	$\{10, 11\}$	11 on $[1, 27]$, 4 on $[30, 76]$	44.0

Table 2.8: Policies and computation times for three different changeover times. The "Short" column indicates configuration pairs requiring a single time interval for a changeover, whereas the pairs in the "Long" column require three time intervals.

2.4 Optimizing Over a Metroplex of Airports

We now extend the problem to consider multiple airports operating within close proximity of one another. This is often referred to as a *metroplex* of airports. In this case it is not sufficient, in general, to optimize each airport separately, since one may end up with an infeasible solution for the system as a whole due to the interactions between arrivals and departures at the different airports. This relationship has been studied in [11], [10] and [36]. In particular, the capacity of a metroplex as a system will in general be lower than the sum of the capacities of its individual airports.

Furthermore, the use of a given configuration at one airport will often impact the range of configurations which can be used simultaneously at neighboring airports within the metroplex. Consider for example the New York metroplex consisting of the airports of Newark (EWR), Kennedy (JFK), LaGuardia (LGA), Islip (ISP) and Teterboro (TEB). One of the many instances of the interactions mentioned above is that when JFK operates with landings on Runway 13L in IMC, LGA must also use its Runway 13 for landings and must also coordinate departures, on either its Runway 4 or its Runway 13 with JFK [31].

2.4.1 Multiple Airport Mixed Integer Programming Model

In extending the mixed integer programming model to this case, an extra index is added to each variable, corresponding to the relevant airport $p \in \mathcal{P}$, and the constraints and definitions of sets and parameters are modified accordingly. In addition, we return to the original assumption of constant changeover times across all airports. The resulting mixed integer program is presented below, and can be viewed as a larger version of the single airport case, with several coupling constraints added.

$$
\begin{aligned}
\textbf{P2-4:}\quad \min\quad & \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}}(c_{pt}u_{pt}+q_{pt}v_{pt}) \\
\text{s.t.}\quad & u_{pt}-u_{p,t-1}+\sum_{k\in\mathcal{K}_{pt}} y_{pkt}=a_{pt},\quad \forall p\in\mathcal{P},\forall t\in\mathcal{T}, && \text{(2.8a)}\\
& v_{pt}-v_{p,t-1}+\sum_{k\in\mathcal{K}_{pt}} x_{pkt}=d_{pt},\quad \forall p\in\mathcal{P},\forall t\in\mathcal{T}, && \text{(2.8b)}\\
& \gamma_{pjk}y_{pkt}-\beta_{pjk}x_{pkt}-\alpha_{pjk}z_{pkt}\le 0,\\
& \qquad\qquad \forall p\in\mathcal{P},\forall j\in\mathcal{J}^t_{pk},\forall k\in\mathcal{K}_{pt},\forall t\in\mathcal{T}, && \text{(2.8c)}\\
& \gamma'_{pjt}\sum_{k\in\mathcal{K}_{pt}} y_{pkt}-\beta'_{pjt}\sum_{k'\in\mathcal{K}_{pt}} x_{pk't}\le\alpha'_{pjt},\quad \forall p\in\mathcal{P},\forall j\in\mathcal{J}'_{pt},\forall t\in\mathcal{T}, \\
& && \text{(2.8d)}\\
& \sum_{k\in\mathcal{K}_{pt}} z_{pkt}\le 1,\quad \forall p\in\mathcal{P},\forall t\in\mathcal{T}, && \text{(2.8e)}\\
& z_{pkt}+\sum_{k'\in\mathcal{K}_{pt-1}\setminus\{k\}} z_{pk',t-1}\le 1,\quad \forall p\in\mathcal{P},\forall k\in\mathcal{K}_{pt},\forall t\in\mathcal{T}\setminus\{1\},\\
& && \text{(2.8f)}\\
& z_{pk,t-1}+\sum_{k'\in\mathcal{K}_{pt}\setminus\{k\}} z_{pk't}\le 1,\quad \forall p\in\mathcal{P},\forall k\in\mathcal{K}_{pt-1},\forall t\in\mathcal{T}\setminus\{1\},\\
& && \text{(2.8g)}\\
& z_{pkt}+z_{p'k't}\le 1,\quad \forall p\in\mathcal{P},\forall k\in\mathcal{K}_{pt},\forall\{p',k'\}\in\mathcal{N}_{pk},\forall t\in\mathcal{T}, && \text{(2.8h)}\\
& \bar{\gamma}^i_j\sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}_{pt}} y_{pkt}-\bar{\beta}^i_j\sum_{p'\in\mathcal{P}}\sum_{k'\in\mathcal{K}_{p't}} x_{p'k't}\le\bar{\alpha}^i_j,\quad \forall i\in\mathcal{I},\forall j\in\bar{\mathcal{J}}_i,\forall t\in\mathcal{T},\\
& && \text{(2.8i)}\\
& z_{pkt}\in\{0,1\},\quad \forall p\in\mathcal{P},\forall k\in\mathcal{K}_{pt},\forall t\in\mathcal{T}, && \text{(2.8j)}\\
& u_{pt},v_{pt}\in\mathbb{Z}_+,\quad \forall p\in\mathcal{P},\forall t\in\mathcal{T}, && \text{(2.8k)}\\
& y_{pkt}\ \ge\ 0,\quad \forall p\in\mathcal{P},\forall k\in\mathcal{K}_{pt},\forall t\in\mathcal{T}, && \text{(2.8l)}\\
& x_{pkt}\ \ge\ 0,\quad \forall p\in\mathcal{P},\forall k\in\mathcal{K}_{pt},\forall t\in\mathcal{T}, && \text{(2.8m)}
\end{aligned}
$$

$$u_{p0}, v_{p0} = 0, \quad \forall p \in \mathcal{P}. \tag{2.8n}$$

Addressing first the similarities of P2-4 with P2-1, our objective is again to minimize the total cost of delays incurred, where u_{pt} and v_{pt} are the number of arrivals and departures, respectively, which go unserved at airport p at time t. These variables are defined by Constraints (2.8a), (2.8b), (2.8k) and (2.8n). Constraints (2.8c) force the operating point, (x_{pkt}, y_{pkt}), given that we operate in the $(p, k)^{th}$ RCCE at time t, to lie within this RCCE, while forcing it to zero if $z_{pkt} = 0$, in which case we do not operate in RCCE (p, k) at time t. Constraints (2.8e) state that, at any time t, we may operate in at most one configuration at each airport, while Constraints (2.8f) model the cost of changing from one configuration to another at each airport by enforcing a delay of one time interval. Finally, for a fixed airport p, Inequalities (2.8d) are identical to Inequalities (2.3) which were added to P2-1, and similarly, Inequalities (2.8g) are the same as Inequalities (2.2), overlooking the notational difference of an extra p in the subscripts.

In addition to these constraints, which are in essence the same in P2-4 as in P2-1, we model the interdependence between airports in Constraints (2.8h) and (2.8i) as follows. Letting $\mathcal{N}_{pk} = \{\{p'_1, k'_1\}, \{p'_2, k'_2\}, \ldots\}$ be the set of pairs such that we cannot operate in configuration k at airport p while also operating in configuration k' at airport p', for any $\{p', k'\} \in \mathcal{N}_{pk}$, Constraints (2.8h) state that we cannot operate in configurations at different airports which are incompatible. In Constraints (2.8i), we assume that for each index i in some set $\mathcal{I}$, we have a system-wide capacity envelope in effect at every time interval which models the extra dependence between airports, and let $\bar{\alpha}^i_j$, $\bar{\beta}^i_j$ and $\bar{\gamma}^i_j$ be the parameters of the j^{th} piece of the i^{th} capacity envelope, where $j \in \bar{\mathcal{J}}_i$. Indeed, the dependence may be much deeper than this, ideally requiring capacity envelopes for every combination of airports and configurations. However, such envelopes may be difficult to obtain, requiring sophisticated computational modeling of the system, whereas the system-wide envelopes above might be obtained through observed system data.

Configurations Unavailable[†]	**RCCE Unavailable**	**Some {{p,k}, {p′, k′}} Disallowed**[‡]	**Optimal Value**	**Time (s)**
No	No	No	28422	2.6
Yes	No	No	35120	23.5
Yes	Yes	No	36424	79.6
No	No	Yes	28422	1.3
Yes	No	Yes	35120	27.3
Yes	Yes	Yes	36862	30.3

Table 2.9: Effect of problem characteristics on computation time for the metroplex case with no system-wide RCCE. [†]$(1, 4, 13)$, $(1, 9, 20)$, $(2, 2, 13)$, $(2, 4, 20)$, $(3, 3, 13)$, $(4, 2, 5)$, $(4, 2, 6)$ [‡]$\{\{1, 9\}, \{2, 3\}\}$, $\{\{2, 3\}, \{4, 4\}\}$, $\{\{3, 2\}, \{5, 4\}\}$

2.4.2 Computational Results

We show here a large example with 5 airports, which have, respectively, 10, 8, 4, 4 and 4 configurations and 26, 20, 11, 10 and 12 RCCE, giving a total of 30 configurations and 79 RCCE. Problem characteristics are varied, and computational results are presented in Tables 2.9 and 2.10.

In Table 2.9, when system-wide capacity envelopes, Constraints (2.8i), are omitted, one can observe that the solution times are highly variable, depending on the characteristics of the problem. In varying the characteristics, we start by solving the full problem (i.e., with all configurations and RCCE available and all combinations allowed), and then modify the problem so that the optimal solution becomes infeasible. Proceeding in this manner, one can see that disallowing certain combinations of configurations being used between airports reduces computation time, while removing the availability of certain RCCE and configurations at certain times can greatly increase computation time.

These trends make sense, since disallowing combinations of configurations between airports reduces problem size, while progressively removing optimal solutions by changing the problem data is likely to result in an optimal solution which is more difficult to obtain by branch and bound. Note as well that each of the first three rows of Table 2.9 closely corresponds to solving five single airport problems, since there is

Configurations Unavailable[†]	**RCCE Unavailable**	**Some $\{\{p,k\}, \{p', k'\}\}$ Disallowed[‡]**	**Bounds on Optimal Value**	**Optimality Gap (%)**
No	No	No	$(83397, 83904)$	0.6
Yes	No	No	$(83397, 85744)$	2.7
Yes	Yes	No	$(83397, 84342)$	1.1
No	No	Yes	$(83397, 83896)$	0.6
Yes	No	Yes	$(83397, 85206)$	2.1
Yes	Yes	Yes	$(83397, 84220)$	1.0

Table 2.10: Effect of problem characteristics on the bounds on the optimal objective value obtained within 5 minutes for the metroplex case with a system-wide RCCE. [†]$(1, 4, 13)$, $(1, 9, 20)$, $(2, 2, 13)$, $(2, 4, 20)$, $(3, 3, 13)$, $(4, 2, 5)$, $(4, 2, 6)$ [‡]$\{\{1, 9\}, \{2, 3\}\}$, $\{\{2, 3\}, \{4, 4\}\}$, $\{\{3, 2\}, \{5, 4\}\}$

no coupling between the five airports in these instances. This gives further evidence that the models we have presented to solve the single airport problem are tractable.

Table 2.10 shows computational results once a system-wide RCCE, Constraint (2.8i), has been added. First, one should note that no optimal solutions were obtained within 5 minutes, and hence we present the bounds obtained. This is clearly a significantly worsened performance, compared to the single-airport case. However, it is encouraging that good feasible solutions were always obtained within one minute, and that the best solutions obtained after 5 minutes were typically within about 2% of optimality. In addition, observe that in this example the system-wide RCCE is quite restrictive given the increase in the optimal objective values that can be inferred from comparing Tables 2.9 and 2.10.

It is evident that the addition of a system-wide RCCE has a significant effect on computation time and as a result the metroplex formulation is less effective than the single airport formulation. Nevertheless, good solutions are obtained within a practical amount of time.

2.5 Further Extensions

We consider next extensions to the basic model that address issues occasionally arising in practice. Suppose that there are certain environmental constraints on certain configurations, such as federal or local regulations (or, in many cases, local “letters of understanding”) governing airport operations. For example, the use of a configuration which leads to departures taking off over a residential area may be prohibited or discouraged during certain times of the day. We show here that the model can easily be extended to accommodate such restrictions.

Consider, for example, the following cases for the single airport problem:

1. The maximum total operating time in configuration k is S_1 time intervals in any continuous period of length $S_2 > S_1$. We add the following $T + 1 - S_2$ constraints:

$$\sum_{t \in \mathcal{Q}} z_{kt} \leq S_1, \quad \forall t' \in \{1, 2, \ldots, T + 1 - S_2\},$$
$$\text{where } \mathcal{Q} = \{t', t' + 1, \ldots, t' + S_2 - 1\} \subset \mathcal{T}. \tag{2.9}$$

2. Once we operate in configuration k, and then stop, we may not resume operation in this configuration until it has been inoperative for S time intervals. Assuming that there are at least S time intervals remaining after some time t, we want $z_{kt} = 1, z_{k,t+1} = 0 \ \Rightarrow \ z_{k,t+2}, z_{k,t+3}, \ldots, z_{k,t+S} = 0$.

   We add $2T - 4$ variables, w_t and ψ_t, and the following $3T - 6 + (T - S)(S - 1) + (S - 2)(S - 1)/2$ constraints to our model:

$$w_t \geq 0, \quad \forall t \in \{1, 2, \ldots, T - 2\}, \tag{2.10a}$$

$$w_t \geq z_{kt} - z_{k,t+1}, \quad \forall t \in \{1, 2, \ldots, T - 2\}, \tag{2.10b}$$

$$\psi_t \geq z_{k,t+j+1}, \quad \forall t \in \{1, 2, \ldots, T - 2\}, \forall j \in \{1, 2, \ldots, s'\},$$
$$\text{where } s' = \min\{S - 1, T - t - 1\}, \tag{2.10c}$$

$$\psi_t + w_t \leq 1, \quad \forall t \in \{1, 2, \ldots, T - 2\}. \tag{2.10d}$$

Note from Constraints (2.10b) that $w_t \geq 1$ if we operate in the k^{th} configuration at time t but not at time $t+1$, and from Constraints (2.10c) that $\psi_t \geq 1$ if we operate in configuration k in any of the s' time intervals after interval $t+1$. Constraints (2.10d) then state that both of these events cannot occur.

Given that such extensions would normally complicate the decision-making process significantly, the ease with which the MIP models can incorporate them is an important benefit.

2.6 Summary

In this chapter, we have presented a strong mixed integer programming formulation to solve the single airport RCM and ADRB problems. Evidence provided in the form of computational results on problems of realistic magnitude indicates that our model solves quickly enough to be implemented in a real world setting. In order to obtain an estimate of the expected benefits from our optimization approach, a sophisticated optimization-based heuristic was developed which reveals the potential cost savings to be indeed significant. We have also shown that a number of additional potential constraints and local considerations can be incorporated into the models with little difficulty. Finally, we have also proposed an extension of the model to optimize over a metroplex of airports.

Chapter 3

Tactical Optimization of Air Traffic at Airports: A Unified Approach

In this chapter[1], we seek to optimize the overall airport surface and near terminal area operations problem, involving the following key decisions:

a) selecting a runway configuration sequence, i.e., determining which runways are open at which times and whether they will process arrivals and/or departures;

b) assigning flights to runways and determining the sequence in which flights are processed at each runway (i.e., when they take off or land);

c) determining the gate-holding duration of departures and speed-control of arrivals outside of the near-terminal airspace, if any;

d) routing flights to their assigned runway at the desired time and onwards within the terminal area and the near-terminal airspace.

In Chapter 2 we solved the airport runway configuration management (RCM) problem (a) above, and the arrival/departure runway balancing (ADRB) problem in

[1] The Massachusetts Institute of Technology filed a patent application related to this work on July 5, 2011 entitled “Airport Operations Optimization” by Dimitris J. Bertsimas and Michael J. Frankovich, United States of America Serial No. 13/176033. Any inquiries regarding the technology or licensing the patent can be directed to the M.I.T. Technology Licensing Office, http://web.mit.edu/tlo.

a single optimization model, as well as proposing an extension to the case of airports in a metroplex with shared airspace. This work was more strategic in nature to that of this chapter in that it presents no directive for controllers to achieve the desired balance of arrivals and departures to be served at any moment, in terms of specific flight assignments. Furthermore, its reliance on the heavy machinery of RCCE may be problematic, not only due to the difficulty in obtaining them, but also because they represent the average maximum throughput possible for each runway configuration, ignoring that the capacity of a configuration may vary from time to time depending for example on the sequence of different aircraft types at each runway. In this chapter, capacity is modeled using much more fundamental units, resulting in greater accuracy. For example, we take as inputs the travel speeds of aircraft, the required separation between aircraft, and the structure of the taxiway system and near-terminal airspace, which all go towards determining a more precise, and time-varying, maximal throughput.

3.1 Introduction

The aviation and optimization communities have generated extensive literature on the various subproblems of airport optimization outlined above. Some key references were described in the Introduction, but those studies have focused mainly on a single subproblem at a time in isolation ([17], [34], [44], [2], [41] on runway sequencing, [21], [35], [13] [12], [40], [30], [38], [29] on surface management), with the notable exceptions of [25] and [15], which merged the runway sequencing and taxiing problems. In this chapter, we present what is to the best of our knowledge the first truly *unified and tractable optimization approach* to solve the overall ATFM problem at a single airport. That is, the first optimization approach which solves subproblems (a) – (d) above together such that a (near-) system-optimal solution is attained within several minutes. The model is a general one – applicable to any airport, regardless of the runway, taxiway, or airspace design. This is a significant contribution due to both the size of the problem and the complexity of its subproblems, notably the runway sequencing subproblem. As a result of these characteristics, a naïve attempt to solve this overall problem would be far from computationally tractable, and it is only through our use of appropriate modeling that we have been able to overcome this tractability challenge. Furthermore, solving the individual subproblems in isolation using the existing literature may lead to overall solutions which are grossly sub-optimal, or indeed infeasible.

The most notable aspects of our modeling approach are that:

i) Our definition of decision variables for runway sequencing leads to a greatly reduced state-space. This is achieved by capitalizing on the fact that the minimum separation required between two flights at a single runway depends only on the weight-class category and orientation of the flights involved (i.e. heavy arrival, small departure, etc), and as a result we do not need such variables for every flight, but rather only for each flight type.

ii) We break the problem down into two natural stages of optimization, which both increases tractability and only very slightly affects optimality.

We present extensive computational experience using real-world datasets for two international airports, Boston Logan International (BOS) and Dallas/Fort Worth (DFW), which weighs in significant evidence to support firstly the claim of computational tractability, and secondly the claim that our optimization can provide significant benefits for air traffic systems.

Outline of Chapter

This chapter is structured as follows:

In Section 3.2, we present Phase One of our two-phase approach to optimizing the entirety of key air traffic flow management decisions to be made at an airport and within its near-terminal airspace. Under a certain assumption, the Phase One solution is a complete one.

In Section 3.3, we present Phase Two of the methodology, which uses the solution from Phase One to tractably solve the problem under an assumption which is much milder than that of Phase One, and which is indeed a very realistic one.

In Section 3.4, we present extensive computational experience based on real-world historic datasets at DFW and BOS to demonstrate both that our methodology is computationally tractable, and that it can present significant benefits in practice.

3.2 The Airport Operations Optimization Problem

In this section we present a novel binary optimization model which represents Phase One of our two-phase approach to solve the entirety of key air traffic flow management decisions to be made at an airport and within its near-terminal airspace. We shall call this the *airport operations optimization problem* (AOOP). The AOOP can be characterized by the set of decisions to be made, which comprises assigning for every departure:

i) a pushback time (and hence a gate-holding time);

ii) a runway assignment and departure fix assignment;

iii) a route from gate to assigned runway, and then to departure fix, with timing;

and vice-versa for every arrival:

iv) a time at arrival fix (which may imply a speed control policy before reaching the fix);

v) a runway assignment and gate assignment;

vi) a route from arrival fix to assigned runway, and then to gate, with timing.

We now provide a high-level description of our two-phase approach to solving the AOOP, as well as the corresponding motivation. The capacitated elements of the near-terminal area are: 1) the gates, 2) the taxiways, 3) the runways, and 4) the near-terminal airspace. Our approach focuses initially (in Phase One) on the runway capacities since it is our view that these present the key bottleneck of the system, and assumes that the gate, taxiway and near-terminal airspace capacities are non-binding. Under this assumption, the solution obtained in Phase One is a complete one – optimal for the AOOP.

Realizing that this assumption may not be entirely realistic in practice, we then relax the assumption and make use of the phase-one solution to form a second-phase optimization problem which is relatively easy to solve. The solution to this second phase of optimization is guaranteed to be feasible for the AOOP, provided flight deadlines are not hard, which is the case in practice, and which we shall assume throughout this thesis. Another way we can view our two-phase approach is that in Phase One we obtain the part of our solution corresponding to subproblems (a) and (b), while in Phase Two we obtain the part corresponding to subproblems (c) and (d).

It is in our particular decomposition of the AOOP into these two natural and complimentary phases that much of our contribution lies. As will be shown, it greatly increases computational tractability without a significant sacrifice in optimality. Based on our belief mentioned above about the nature of airport capacity, we might expect the solution obtained from the second phase to be in general very similar to that of Phase One, and hence very close to optimal. Indeed, the computational experience with real-world data in Section 3.4 will show that there is almost no loss of optimality in the real-world instances to which we apply our methodology.

3.2.1 Data

Now we detail the data requirements of this first stage optimization problem. We consider a time horizon $\mathcal{T} = \{1, \ldots, T\}$ of approximately one hour, discretized into small intervals of 20 seconds in length, being small enough so that proper separation times can be achieved. We have a set of flights $\mathcal{F}$, with each flight having a weight class w (heavy, large, small, or Boeing-757) and an orientation o (arrival or departure). The pair $\boldsymbol{i} = (w, o)$ will be referred to as a *flight type*, belonging to the set of flight types $\mathcal{C}$. Flight types are defined in this way since the minimum separation time required between two flights on the same runway will depend on these characteristics. There is also a set of runway configurations $\mathcal{K}$. Each configuration k is described by a set of pairs $\{(r_1, m_1), \ldots, (r_N, m_N)\}$, a pair comprising a runway r and a mode of operation m (i.e., arrivals only, departures only, or mixed mode).

The following is a complete list of the data:

$\mathcal{T}$ = $\{1, \ldots, T\}$ = the set of time intervals comprising the time horizon considered;

$\mathcal{C}$ = the set of flight types, each of which is a *pair* $\boldsymbol{i} = (w, o)$ corresponding to a weight class category w and a flight orientation (arrival/ departure) o;

$\mathcal{C}_A, \mathcal{C}_D$ = the set of flight types whose orientation is arrival, departure, respectively;

$\mathcal{F}$ = $\mathcal{F}_A \cup \mathcal{F}_D = \cup_{\boldsymbol{i} \in \mathcal{C}} \mathcal{F}_{\boldsymbol{i}}$ = the set of flights;

$\mathcal{R}$ = the set of runways, each of which includes a single, fixed, direction of operation;

$\mathcal{R}_f, \mathcal{R}_{\boldsymbol{i}}$ = the set of runways which is feasible for flight f, or for some flight of type $\boldsymbol{i}$, respectively. The feasibility of a given runway for a given flight depends on several factors, notably aircraft type and runway dimensions;

$\mathcal{V}$ = the set of pairs of runways $\left\{(r_1^1, r_2^1), \ldots, (r_1^N, r_2^N)\right\}$ where pairwise separation must be enforced, for example intersecting or close parallel runways;

$\mathcal{K}$ = the set of runway configurations, each of which is a *set of pairs* $\boldsymbol{k} = \{(r_1, m_1), \ldots, (r_N, m_N)\}$, where m_j is the mode of operation of runway r_j. The operating mode can be arrivals only, departures only, or "mixed mode," in which both arrivals and departures can be processed simultaneously;

$\mathcal{R}_{\boldsymbol{k}}$ = the set of runways used by configuration $\boldsymbol{k}$;

$\mathcal{I}_{r\boldsymbol{k}}$ = the set of flight types that can use runway r under configuration $\boldsymbol{k}$;

$\mathcal{U}_t$ = the set of runways which cannot be used at time t due, for example, to strong crosswinds or tailwinds;

$\mathcal{T}_r^f$ = $\{\underline{T}_r^f, \underline{T}_r^f + 1, \ldots, \overline{T}_r^f\}$ = the set of feasible times for flight f to arrive at runway r, considering the flight's starting time and location and the shortest paths to and from r, when unimpeded by traffic;

$\underline{T}_{o_f}^f$ = the earliest possible release time of flight f from its origin within the system (i.e., from a gate or arrival fix);

$l_{\boldsymbol{i}}^r$ = the number of time intervals constituting the runway occupancy time of flights of type $\boldsymbol{i}$ at runway r;

$s_{\boldsymbol{ij}}^r$ = the minimum number of time intervals of separation required between aircraft when an aircraft of type $\boldsymbol{j}$ follows an aircraft of type $\boldsymbol{i}$ at runway r, the calculation of which is outlined in Appendix A. We refer the reader to [16] for more details, but we note that this is always at least equal to $l_{\boldsymbol{i}}^r$, the runway occupancy time of the first aircraft;

$s_{\boldsymbol{ij}}^{(r,r')}$ = the minimum number of time intervals of separation required at intersecting/closely-spaced parallel runways when an aircraft of type $\boldsymbol{j}$ scheduled at runway r' follows an aircraft of type $\boldsymbol{i}$ scheduled at runway r, if $(r, r') \in \mathcal{V}$;

β_A^A, β_D^A = constants weighting the delay cost in the air for arrivals and departures, respectively, with $\beta_A^A > \beta_D^A$;

β_A^G, β_D^G = constants weighting the delay cost on the ground for arrivals and departures, respectively;

β_G = a constant weighting the delay cost at the gate before pushback, for departures;

d_r^f = the distance of a shortest path for flight f from runway r to its destination, which is either a departure fix or gate;

K = a large constant which penalizes each configuration changeover.

3.2.2 Decision Variables

We define the following binary decision variables for our model:

$$\omega_{\boldsymbol{k}t} = \begin{cases} 1, & \text{if configuration } \boldsymbol{k} \text{ is active at time } t, \\ 0, & \text{otherwise;} \end{cases}$$

$$\varphi_r^f = \begin{cases} 1, & \text{if flight } f \text{ is assigned to runway } r, \\ 0, & \text{otherwise;} \end{cases}$$

$$\psi_{rt}^{\boldsymbol{i}} = \begin{cases} 1, & \text{if a flight of type } \boldsymbol{i} \text{ arrives at runway } r \text{ at time } t, \\ 0, & \text{otherwise;} \end{cases}$$

$$\chi_t = \begin{cases} 1, & \text{if a change of configuration occurs at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

We note that one of the key ideas behind this model and its tractability is that we have chosen to define the variables $\boldsymbol{\psi}$ by *flight type*, rather than by flight, capitalizing on the fact that separation depends only on flight type, and greatly reducing the number of variables to $O(|\mathcal{C}||\mathcal{R}||\mathcal{T}| + |\mathcal{F}||\mathcal{R}|)$, rather than $O(|\mathcal{F}||\mathcal{R}||\mathcal{T}|)$, and the number of constraints to $O(|\mathcal{C}|^2|\mathcal{R}||\mathcal{T}|)$, rather than $O(|\mathcal{F}|^2|\mathcal{R}||\mathcal{T}|)$. Indeed, this modeling technique may be applied for general bounded-TSP type problems.

3.2.3 Objective Function

Our objective is to minimize the function (3.1) below, which represents a weighted summation of flight delay costs:

$$\begin{aligned}\Psi \triangleq \sum_{\boldsymbol{i}\in\mathcal{C}} \left(\beta_D^G.\mathbb{I}_{\{\boldsymbol{i}\in\mathcal{C}_D\}} + \beta_A^A.\mathbb{I}_{\{\boldsymbol{i}\in\mathcal{C}_A\}}\right) \sum_{r\in\mathcal{R}_{\boldsymbol{i}}} \sum_{t\in\mathcal{T}} t\psi_{rt}^{\boldsymbol{i}} - \sum_{f\in\mathcal{F}} \left(\beta_D^G.\mathbb{I}_{\{f\in\mathcal{F}_D\}} + \beta_A^A.\mathbb{I}_{\{f\in\mathcal{F}_A\}}\right) \underline{T}_{o_f}^f \\ + \sum_{f\in\mathcal{F}} \sum_{r\in\mathcal{R}_f} \left(\beta_D^A.\mathbb{I}_{\{f\in\mathcal{F}_D\}} + \beta_A^G.\mathbb{I}_{\{f\in\mathcal{F}_A\}}\right) \cdot d_r^f \varphi_r^f \\ - \left(\beta_D^G - \beta_G\right) \cdot \left(\sum_{\boldsymbol{i}\in\mathcal{C}_D} \sum_{r\in\mathcal{R}_{\boldsymbol{i}}} \sum_{t\in\mathcal{T}} t\psi_{rt}^{\boldsymbol{i}} - \sum_{f\in\mathcal{F}_D} \sum_{r\in\mathcal{R}_f} \underline{T}_r^f \varphi_r^f \right) \\ + K \cdot \sum_{t\in\mathcal{T}} \chi_t. \qquad (3.1)\end{aligned}$$

Note that here, we introduce the notation $\mathbb{I}_{\{statement\}}$ to represent the indicator variable which is equal to 1 if the statement is true, and 0 if the statement is false.

This can be summarized as a summation over all flights of the following terms:

$$\begin{pmatrix}\text{weighted time from}\\ \text{first time period}\\ \text{until touchdown/takeoff}\end{pmatrix} - \begin{pmatrix}\text{weighted time from}\\ \text{first time period}\\ \text{until start time}\end{pmatrix} + \\ \begin{pmatrix}\text{weighted time}\\ \text{from touchdown/takeoff}\\ \text{to destination}\end{pmatrix} - \begin{pmatrix}\text{weighted}\\ \text{gate-holding}\\ \text{duration}\end{pmatrix} + \begin{pmatrix}\text{configuration}\\ \text{change}\\ \text{penalty}\end{pmatrix}.$$

In more detail, the first line gives the weighted sum over all flights of the length of time from a flight's starting time until its touchdown/takeoff time. The coefficients weight the cost of delays in the air appropriately. The next line penalizes the use of different runways, adding the length of the weighted shortest path for flight f from its assigned runway r to its gate, $\beta_A^G d_r^f$, for arrivals and to its departure fix, $\beta_D^A d_r^f$, for departures. In the third line we make the assumption that if departures do not arrive at their assigned runway at the earliest possible time, then they are held at the gate for the slack duration, and so we adjust the objective function accordingly. The final term of the objective function penalizes each change of configuration, preventing too many from occurring, as this is undesirable in practice.

3.2.4 The Binary Optimization Problem

A binary optimization problem for the AOOP under non-binding network capacity is then the following:

P3-1: min $\quad \Psi$

$$\text{s.t.} \quad \sum_{\boldsymbol{k}\in\mathcal{K}} \omega_{\boldsymbol{k}t} = 1, \quad \forall t \in \mathcal{T}, \tag{3.2a}$$

$$\psi_{rt}^{\boldsymbol{i}} = 0, \quad \forall \boldsymbol{i} \in \mathcal{C},\ r \in \mathcal{U}_t,\ t \in \mathcal{T}, \tag{3.2b}$$

$$\psi^{\boldsymbol{i}}_{r,t-h} + \psi^{\boldsymbol{j}}_{rt} \leq 1, \quad \forall \boldsymbol{i}, \boldsymbol{j} \in \mathcal{C},\, r \in \mathcal{R}_{\boldsymbol{i}} \cap \mathcal{R}_{\boldsymbol{j}},$$

$$h \in \{1, \ldots, \min\{s^{r}_{\boldsymbol{ij}} - 1, t-1\}\},\, t \in \mathcal{T} \setminus \{1\}, \tag{3.2c}$$

$$\psi^{\boldsymbol{i}}_{r,t-h} + \psi^{\boldsymbol{j}}_{r',t} \leq 1, \quad \forall \boldsymbol{i}, \boldsymbol{j} \in \mathcal{C},\, (r, r') \in (\mathcal{R}_{\boldsymbol{i}} \times \mathcal{R}_{\boldsymbol{j}}) \cap \mathcal{V},$$

$$h \in \{0, \ldots, \min\{s^{(r,r')}_{\boldsymbol{ij}} - 1, t-1\}\},\, t \in \mathcal{T}, \tag{3.2d}$$

$$\sum_{\boldsymbol{i} \in \mathcal{C} \,:\, r \in \mathcal{R}_{\boldsymbol{i}}} \psi^{\boldsymbol{i}}_{rt} \leq 1, \quad \forall r \in \mathcal{R},\, t \in \mathcal{T}, \tag{3.2e}$$

$$\psi^{\boldsymbol{i}}_{rt} + \omega_{\boldsymbol{k}t} \leq 1, \quad \forall t \in \mathcal{T},\, \boldsymbol{k} \in \mathcal{K},\, r \in \mathcal{R}_{\boldsymbol{k}}, \boldsymbol{i} \in \bar{\mathcal{I}}_{r\boldsymbol{k}} \,:\, r \in \mathcal{R}_{\boldsymbol{i}}, \tag{3.2f}$$

$$\psi^{\boldsymbol{i}}_{rt} - \sum_{\substack{\boldsymbol{k} \in \mathcal{K} \,:\, r \in \mathcal{R}_{\boldsymbol{k}}, \\ \boldsymbol{i} \in \mathcal{I}_{r\boldsymbol{k}}}} \omega_{\boldsymbol{k},t+h} \leq 0, \quad \forall \boldsymbol{i} \in \mathcal{C},\, r \in \mathcal{R}_{\boldsymbol{i}},$$

$$h \in \{0, \ldots, \min\{l^{r}_{\boldsymbol{i}} - 1, T - t\}\},\, t \in \mathcal{T}, \tag{3.2g}$$

$$\sum_{r \in \mathcal{R}_f} \varphi^{f}_{r} = 1, \quad \forall f \in \mathcal{F}, \tag{3.2h}$$

$$\sum_{\substack{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f, \\ t \geq \overline{T}^{f}_{r} - l^{r}_{\boldsymbol{i}} + 1}} \varphi^{f}_{r} \leq \sum_{\tau=1}^{t} \psi^{\boldsymbol{i}}_{r\tau} \leq \sum_{\substack{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f, \\ t \geq \underline{T}^{f}_{r}}} \varphi^{f}_{r}, \quad \forall \boldsymbol{i} \in \mathcal{C},\, r \in \mathcal{R}_{\boldsymbol{i}},\, t \in \mathcal{T}, \tag{3.2i}$$

$$\chi_t - \omega_{\boldsymbol{k}t} + \omega_{\boldsymbol{k},t-1} \geq 0, \quad \forall \boldsymbol{k} \in \mathcal{K},\, t \in \mathcal{T} \setminus \{1\}, \tag{3.2j}$$

$$\omega_{\boldsymbol{k}t} \in \{0, 1\}, \quad \forall \boldsymbol{k} \in \mathcal{K},\, t \in \mathcal{T}, \tag{3.2k}$$

$$\varphi^{f}_{r} \in \{0, 1\}, \quad \forall f \in \mathcal{F},\, r \in \mathcal{R}_f, \tag{3.2l}$$

$$\psi^{\boldsymbol{i}}_{rt} \in \{0, 1\}, \quad \forall \boldsymbol{i} \in \mathcal{C},\, r \in \mathcal{R}_{\boldsymbol{i}},\, t \in \mathcal{T}, \tag{3.2m}$$

$$\chi_t \in \{0, 1\}. \quad \forall t \in \mathcal{T}. \tag{3.2n}$$

Constraints (3.2a) require *exactly* one configuration to be used at any time, while Constraints (3.2b) prevent flights from occupying runways which are not available at time t. Note that even if a runway is not available at a given time, a configuration may be used (as indicated by the $\boldsymbol{\omega}$ variables) which uses that runway, and its capacity is set to zero by the latter set of constraints, rather than by the former. This method of controlling runway and configuration availability leads to fewer configurations being required in the model (any "sub-configuration" of a configuration does not require additional configuration variables) – see Appendix B for the configurations used at BOS and DFW. In addition, it enables us to add an extra class of valid inequalities (Inequalities (3.3a), detailed in Proposition 3.1) to strengthen the model.

Constraints (3.2c) can generally be referred to as the *separation constraints*, which state that if we process a flight of type $\boldsymbol{i}$, then we must wait at least $s^r_{\boldsymbol{ij}}$ time periods before processing a flight of type $\boldsymbol{j}$, on any given runway. An important point to note here is that these constraints correctly model the fact that the triangle inequality is not respected in this problem. In other words, a sequence of flights $f \rightarrow g \rightarrow h$ may not be legal/safe if we only respect the minimum separations required between flights f and g, and between g and h separately – we also require that the minimum separation between flights f and h be observed.

Constraints (3.2d) enforce a similar separation requirement when we have a pair of runways (r, r') between which separation must be enforced, for example in the case of intersecting or sufficiently close parallel runways. We note that in the computational experience of this chapter, we shall take the (only slightly) conservative approach of modeling close parallel runways as though there is only a single runway, instead of with these additional constraints.

The final consideration regarding the separation between flights is that runway separation alone is not enough – flights also need to be separated throughout the airspace. In calculating our same-runway separation rules, we have incorporated the different flight velocities and their impact on the separation along a common flight path of 5 nautical miles (see Appendix A). When flights use different runways, the relevant separation requirements will be enforced in Phase Two. We also note here that another alternative is to extend the definition of flight type to include a fix.

We now remind the reader that the definition of $\boldsymbol{\psi}$ is such that $\psi^{\boldsymbol{i}}_{rt} = 1$ if, and only if, a flight of type $\boldsymbol{i}$ *arrives* at runway r at time t, and hence such a flight might (and in general, will) actually occupy the runway for more than one time interval, even though this is not directly tracked by our decision variables $\boldsymbol{\psi}$. Then, Constraints (3.2e) state that only one flight may arrive at each runway at any given time. This set of constraints, along with Constraints (3.2c) above, enforce the capacity of each runway to be one at all times (recall $s^r_{\boldsymbol{ij}} \geq l^r_{\boldsymbol{i}}$). Constraints (3.2f) disallow the use of runway r for flights of type $\boldsymbol{i}$ if such use is not allowed under configuration $\boldsymbol{k}$. Constraints (3.2g) state that if we process a flight of type $\boldsymbol{i}$ at a given runway r, then that runway must remain open for at least $l^r_{\boldsymbol{i}}$ time periods, corresponding to the runway occupancy time of flights of type $\boldsymbol{i}$.

Constraints (3.2h) state that every flight must be assigned to some runway. Then, Constraints (3.2i) require each flight f to be processed at one of its feasible runways r after its earliest possible touchdown/takeoff time $\underline{T}^f_r$. The left-hand side is equal to the number of flights assigned to runway r which should have been processed by time t (based on our assumed flexible "deadlines"), and the right-hand side is equal to the maximum number of flights assigned to runway r which could feasibly have arrived at r by time t (recall this is based on shortest paths). So, these constraints state that the number of flights of type $\boldsymbol{i}$ assigned to runway r by time t must fall within this range, for every $t \in \mathcal{T}$. These are the only constraints that link the $\boldsymbol{\psi}$ variables with the $\boldsymbol{\varphi}$ variables.

Finally, Constraints (3.2j) enforce $\chi_t = 1$ if a change of configuration occurs at time t, which happens if, and only if $\exists \boldsymbol{k} : \omega_{\boldsymbol{k}t} = 1$ and $\omega_{\boldsymbol{k},t-1} = 0$. Note that this is equivalent to setting $\chi_t = 1$, since $\boldsymbol{\chi}$ is penalized in the objective function and this is the only constraint on $\boldsymbol{\chi}$.

Remarks on the Model

- A helpful way to think about this model is to first suppose that the network made up of the gates, taxiways and near-terminal area airspace has infinite capacity. In this case, all flights can travel along their shortest paths without obstruction and hence arrive at their assigned runway at their assigned time. Then, P3-1 gives an optimal solution to the AOOP, including the optimal configuration schedule (through $\boldsymbol{\omega}$), the optimal runway assignments ($\boldsymbol{\varphi}$), the optimal sequencing of flights ($\boldsymbol{\psi}$), and implicitly an optimal routing of flights. This routing is such that each flight:

i) spends any slack time waiting at its gate, if the flight is a departure;

ii) travels unimpeded along a shortest path from its origin to its assigned runway;

iii) reaches its assigned runway at its assigned time;

iv) travels unimpeded along a shortest path from its assigned runway to its destination.

- A key feature of our methodology is our particular definition of decision variables. A naïve attempt would define variables φ_{rt}^{f}, being equal to one if flight f were at runway r at time t. This, however, would result in computational intractability as the number of flights and time periods increased, especially due to the number of constraints required to enforce minimum between-flight separation rules. Instead, we note that the between-flight separation depends only on the *type* of two adjacent flights, and not on their unique flight identifiers. Here, the type is characterized by a weight-class category and arrival/departure status. Hence, we define our decision variables for the separation constraints based on flight type, giving $\psi_{rt}^{\boldsymbol{i}} = 1$ if a flight of type $\boldsymbol{i}$ is at runway r at time t. As a result, we have a significant reduction in the number of decision variables and constraints.

- Since the variables $\boldsymbol{\psi}$ are defined by flight type, we have a sequence of "flight type slots" at each runway, instead of having a sequence of flights at each runway. However, through the variables $\boldsymbol{\varphi}$ we also have an assignment of flights to runways, and it is through Constraints (3.2i) that we link these two sets of variables. Indeed, finding flight type slots and then allocating specific flights to these slots has been proposed by [1] and [41]. Inspection of these constraints reveals that there is always at least one sequence of flights corresponding to a solution of P3-1, and that such a sequence can be trivially obtained from the solution.

- In terms of our overall two-phase methodology, we can view Phase One as solving subproblems (a) and (b), in the sense that these components will be retained in the solution obtained at the end of Phase Two. The solution to subproblems (c) and (d) will be found in Phase Two, but will in general be very similar to that found here in Phase One.

3.2.5 Valid Inequalities

We can strengthen the formulation P3-1 by adding certain valid inequalities. Let $\underline{s}^r_{\boldsymbol{j}} = \min_{\boldsymbol{i}\in\mathcal{C}}\{s^r_{\boldsymbol{ij}}\}$ be the minimum possible separation time required between two flights at runway r when the second flight is of type $\boldsymbol{j}$. Also let $\mathcal{X}$ be the set of feasible solutions to P3-1 which have the additional property that $\forall t \in \mathcal{T}\setminus\{1\}$, $\exists \boldsymbol{k} : \chi_t = \omega_{\boldsymbol{k}t} - \omega_{\boldsymbol{k},t-1}$. Since χ_t is penalized in the objective function, any optimal solution has this property. Then we have the following proposition:

Proposition 3.1. *The following inequalities are valid for the polyhedron conv($\mathcal{X}$). Furthermore, they are not valid for the linear relaxation of P3-1.*

$$\chi_t - \omega_{\boldsymbol{k},t-1} + \omega_{\boldsymbol{k}t} \geq 0, \quad \forall \boldsymbol{k} \in \mathcal{K},\, t \in \mathcal{T}\setminus\{1\}, \tag{3.3a}$$

$$\chi_t + \omega_{\boldsymbol{k},t-1} + \omega_{\boldsymbol{k}t} \leq 2, \quad \forall \boldsymbol{k} \in \mathcal{K},\, t \in \mathcal{T}\setminus\{1\}, \tag{3.3b}$$

$$\sum_{\boldsymbol{i}\in\mathcal{C}:\, r\in\mathcal{R}_{\boldsymbol{i}}} \sum_{h=0}^{\min\{\underline{s}^r_{\boldsymbol{i}}-1,T-t\}} \psi^{\boldsymbol{i}}_{r,t+h} \leq 1, \quad \forall r \in \mathcal{R},\, t \in \mathcal{T}. \tag{3.3c}$$

Proof. Let $\mathbf{x} \triangleq (\boldsymbol{\omega}', \boldsymbol{\varphi}', \boldsymbol{\psi}', \boldsymbol{\chi}')' \in \mathcal{X}$ be a feasible solution to P3-1 with the required property.

- *Proof of* (3.3a)*:* Fix $\boldsymbol{k} \in \mathcal{K}$, $t \in \mathcal{T}\setminus\{1\}$. Since we penalize χ_t in the objective function,

  $$\chi_t = 1 \iff \exists \boldsymbol{k'} : \omega_{\boldsymbol{k'},t-1} = 0 \text{ and } \omega_{\boldsymbol{k'}t} = 1.$$

  There are three cases, and in each of them it is clear that $\boldsymbol{x}$ satisfies Inequalities (3.3a):

  1. The runway configuration changes to configuration $\boldsymbol{k}$ from some other

Time	1	2	3	4	5
ω_{1t}	0.5	0.5	0.33	0	0
ω_{2t}	0.5	0.5	0.34	0.5	0.5
ω_{3t}	0	0	0.33	0.5	0.5
χ_t	0	0	0.33	0.17	0
Lower bound on χ_t from Inequalities (3.3a)	0	0	0.17	0.33	0

Table 3.1: Example fractional solution to the linear relaxation of P3-1 ruled out by Inequalities (3.3a).

configuration $\boldsymbol{k'}$ at time t:

$$\omega_{\boldsymbol{k'},t-1} = 1,\, \omega_{\boldsymbol{k'}t} = 0 \text{ and } \omega_{\boldsymbol{k},t-1} = 0,\, \omega_{\boldsymbol{k}t} = 1,$$

giving $\chi_t = 1$ by Constraints (3.2j).

2. The runway configuration changes from configuration $\boldsymbol{k}$ to some other configuration $\boldsymbol{k'}$ at time t:

$$\omega_{\boldsymbol{k},t-1} = 1,\, \omega_{\boldsymbol{k}t} = 0 \text{ and } \omega_{\boldsymbol{k'},t-1} = 0,\, \omega_{\boldsymbol{k'}t} = 1,$$

giving $\chi_t = 1$ by Constraints (3.2j).

3. There is no configuration change at time t:

$$\omega_{\boldsymbol{k},t-1} = \omega_{\boldsymbol{k}t} \in \{0,1\},\ \text{giving } \chi_t = 0 \text{ by Constraints (3.2j).}$$

Now we demonstrate in Table 3.1 a partial solution to the linear relaxation of P3-1. First observe that this fractional solution respects Constraints (3.2a) and (3.2j) of P3-1 (we assume the remainder of this solution is such that it satisfies all other, less relevant, constraints). Next, observe that the solution does not satisfy Inequalities (3.3a). This completes this part of the proof.

- *Proof of* (3.3b)*:* Fix $\boldsymbol{k} \in \mathcal{K}$, $t \in \mathcal{T} \setminus \{1\}$. By our optimality assumption, $0 \le \chi_t \le 1$. So we need to show that for $\boldsymbol{x}$, $\omega_{\boldsymbol{k},t-1} + \omega_{\boldsymbol{k}t} > 1 \implies \chi_t = 0$.

Time	1	2	3	4	5	6
ψ^1_{rt}	0.5	0.5	0.5	0.5	0.5	0.5
ψ^2_{rt}	0	0	0	0	0	0
ψ^3_{rt}	0.5	0.5	0.5	0.5	0.5	0.5

Table 3.2: Example fractional solution ruled out by Inequalities (3.3c). Here r is fixed, there are 3 flight types, and $\min_{\boldsymbol{i},\boldsymbol{j}\in\mathcal{C}} s^r_{\boldsymbol{ij}} = 3$.

Recall $\chi_t > 0 \iff$ we have a configuration change at time t. Hence

$$\begin{aligned}\chi_t > 0 &\iff \{\omega_{\boldsymbol{k},t-1} = \omega_{\boldsymbol{k}t} = 0\} \text{ or } \{\omega_{\boldsymbol{k},t-1} = 0, \omega_{\boldsymbol{k}t} = 1\} \text{ or } \{\omega_{\boldsymbol{k},t-1} = 1, \omega_{\boldsymbol{k}t} = 0\}\\ &\iff \omega_{\boldsymbol{k},t-1} + \omega_{\boldsymbol{k}t} \leq 1.\end{aligned}$$

Now take a feasible solution to the linear relaxation of P3-1 with $\chi_t = 1$ and $\omega_{\boldsymbol{k},t-1} = \omega_{\boldsymbol{k}t} = 0.87$. Clearly this solution does not satisfy Inequalities (3.3b).

- *Proof of* (3.3c)*:* Fix $r \in \mathcal{R}$, $t \in \mathcal{T}$. By Constraints (3.2e) and (3.2c), $\boldsymbol{x}$ has the property that

  $$\forall \boldsymbol{i}, \boldsymbol{j}, \in \mathcal{C} : r \in \mathcal{R}_{\boldsymbol{i}} \cap \mathcal{R}_{\boldsymbol{j}}, \psi^{\boldsymbol{i}}_{rt} = 1 \implies \psi^{\boldsymbol{j}}_{r,t+h} = 0, \forall h \in \{0, \ldots, \min\{s^r_{\boldsymbol{ij}} - 1, T - t\}\}.$$

  So, using the definition of $\underline{s}^r_{\boldsymbol{j}}$, $\boldsymbol{x}$ also has the property that

  $$\forall \boldsymbol{i}, \boldsymbol{j}, \in \mathcal{C} : r \in \mathcal{R}_{\boldsymbol{i}} \cap \mathcal{R}_{\boldsymbol{j}}, \psi^{\boldsymbol{i}}_{rt} = 1 \implies \psi^{\boldsymbol{j}}_{r,t+h} = 0, \forall h \in \{0, \ldots, \min\{\underline{s}^r_{\boldsymbol{j}} - 1, T - t\}\}.$$

  So if $\exists \boldsymbol{i} \in \mathcal{C} : \psi^{\boldsymbol{i}}_{rt} = 1$, then no other term in Inequality (3.3c) can be nonzero. Furthermore, it follows that if $\exists \boldsymbol{i} \in \mathcal{C}$, $h \in \{0, \ldots, \min\{\underline{s}^r_{\boldsymbol{i}} - 1, T - t\}\} : \psi^{\boldsymbol{i}}_{r,t+h} = 1$, then no other term in Inequality (3.3c) can be nonzero. Therefore Inequalities (3.3c) must be satisfied by $\boldsymbol{x}$.

  Table 3.2 shows part of a solution to the linear relaxation of P3-1 which does not satisfy Inequalities (3.3c). First observe that this fractional solution respects Constraints (3.2e) and (3.2c) of P3-1. Then, observe that for this example the left-hand side of Inequalities (3.3c) is $\geq (0.5 + 0.5 + 0.5) + (0 + 0 + 0) + (0.5 + 0.5 + 0.5) = 3 > 1$ for any $t \in \{1, 2, 3, 4\}$.

□

3.3 Phase Two – The Routing Subproblem

In this section we detail the second phase of our optimization approach for the AOOP, addressing the case when the capacity of the gates, taxiways, or airspace becomes binding. This phase can essentially be viewed as the "routing phase," in which we determine a routing of flights to achieve a runway processing schedule which is very close to that obtained in the first phase, if not the same. In particular, we fix the solution from Phase One to subproblems (a) and (b) outlined in Section 3.2 and in this second phase we obtain the solution to subproblems (c) and (d). In more detail, we present a binary optimization problem, P3-2, which takes the solution from P3-1 as an input and outputs a solution to the AOOP which preserves the assignment of flights to runways and the ordering of flights at each runway determined in Phase One, but not necessarily the specific touchdown/takeoff times.

This approach provides the flexibility sufficient to ensure feasibility, provided flight deadlines are not hard, while also ensuring the solution retains the nice properties of the Phase One solution. In the case of infeasibility, we would require the flight deadlines used in the optimization problem P3-2 below to be relaxed, and if necessary, the time horizon increased before re-solving. The approach is informed by our belief that the runways are the most restrictive component of capacity, meaning that there should not be a significant loss of optimality in this second phase. In Section 3.4, we shall support this statement.

3.3.1 Data

In order to model the routing subproblem, the airport network is represented by a directed graph with nodes belonging to the set $\mathcal{S}$, where each node represents a section of taxiway, a runway, an airspace route, a gate, or a fix. For an example of one such simplified network representation, we refer the reader to Appendix C. A list of relevant sets and parameters, building on those of Section 3.2, is given below:

$\mathcal{S}$ = the set of nodes in the airport network;

$\mathcal{S}_f(\subset \mathcal{S})$ = the set of nodes in the airport network feasible for flight f;

$\mathcal{L}_i^f$ = the set of nodes which are successors of node i for flight f;

$\mathcal{P}_i^f$ = the set of nodes which are predecessors of node i for flight f;

$\mathcal{E}_f(\subset \mathcal{S}_f)$ = the set of possible end nodes of flight f;

$\mathcal{T}_i^f$ = $\{\underline{T}_i^f, \underline{T}_i^f + 1, \ldots, \overline{T}_i^f\}$ = the set of feasible times for flight f to arrive at node i, considering the flight's starting time and location and the shortest path to i, when unimpeded by traffic;

$\boldsymbol{c}_f(\in \mathcal{C})$ = the type of flight f;

o_f = initial node of flight f;

l_i^f = the minimum amount of time flight f must spend at node i;

u_{it} = the capacity of node i, in flights, at time t.

3.3.2 Input from Phase One

In addition to the above data, we require several inputs obtained from the Phase One solution. Before we detail these, recall that the solution to P3-1 provides runway assignments for each flight, but only times of flight types at their assigned runways. It does not provide times at which individual flights arrive at their assigned runways (and therefore does not completely specify the flight sequence at each runway) – there is some freedom in assigning specific flights based on the flight type assignments. There is not complete freedom, however. In particular, we can only make swaps amongst flights of the same type which are assigned to the same runway, and only ones which respect the relevant time window constraints. Although one can imagine many possible schemes for determining this ordering, this is not a focus of this chapter and we shall now assume we have fixed such an ordering.

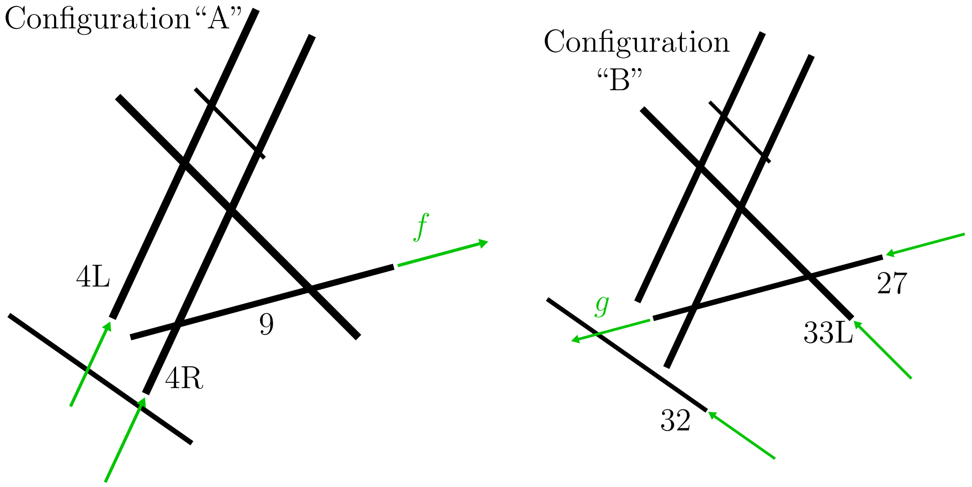

Figure 3-1: Example illustrating an element (f, g) belonging to the set $\mathcal{Q}$ at BOS. The arrows indicate the direction and mode of traffic as dictated by the configuration in use. Suppose in the solution to P3-1 we have: i) configuration A is used first, then configuration B, ii) flight f is assigned to runway 9, and flight g to runway 27. Since runway 27 is not used in configuration A, we have $(f, g) \in \mathcal{Q}$.

$r_f (\in \mathcal{R}_f)$ = the assigned runway node at which flight f should be processed;

$\mathcal{Q}$ = $\{(f_1, g_1), \ldots, (f_k, g_k)\}$ = the set of pairs of flights (f, g) such that the following hold:

i) flight f is scheduled to use runway r in configuration A;

ii) flight g is scheduled to use runway q in configuration B;

iii) configuration A is scheduled for use before configuration B;

iv) runway r is not used in configuration B in a mode of operation that would allow flight f to be processed then;

v) runway q is not used in configuration A in a mode of operation that would allow flight g to be processed then.

Figure 3-1 gives an example of an element of $\mathcal{Q}$. This set will be used to ensure that configuration requirements are respected, since they are not modeled explicitly in the model P3-2 below.

$\mathcal{H}_r \;=\; \{(f_1, f_2), (f_2, f_3), \ldots, (f_{n-1}, f_n)\}$ = the set of pairs of successive flights to be processed on runway r, for each $r \in \mathcal{R}$;

$\mathcal{H}_{(r,r')} \;=\; \{(f_1, f_2), (f_2, f_3), \ldots, (f_{n-1}, f_n)\}$ = the set of pairs of successive flights (f, g) with f being processed on runway r and g being processed on r', for every pair of runways (r, r') at which pairwise separation must be enforced;

$\mathcal{W}_i \;=\; \{(f_1, f_2), (f_2, f_3), \ldots, (f_{n-1}, f_n)\}$ = the set of pairs of flights (f, g) which are processed at *different* runways and which pass through the same fix i in the order $f \to g$ "within close proximity" of each other, and require $s^i_{\boldsymbol{c}(f),\boldsymbol{c}(g)}$ time intervals of separation there.

3.3.3 Decision Variables

We have the following decision variables:

$$z^f_{it} = \begin{cases} 1, & \text{if flight } f \text{ reaches node } i \text{ by time } t, \\ 0, & \text{otherwise;} \end{cases}$$

$$x^f_{it} = \begin{cases} 1, & \text{if flight } f \text{ is at node } i \text{ at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

Note that the $\boldsymbol{z}$ variables are defined as "by" variables in the spirit of [9], which will lead to nice properties in the model formulation.

3.3.4 Objective Function

The objective function which we minimize captures the same quantity as the P3-1 objective function – the weighted sum over the total time it takes for each flight to go through the system, and is described by the function (3.4) below:

$$\Phi \triangleq \sum_{f \in \mathcal{F}_D} \left[\beta_G \left(\gamma^f_{o_f} - \sum_{i \in \mathcal{L}^f_{o_f}} \gamma^f_i - l^f_{o_f} \right) + \right.$$

$$\beta_D^G \left(l_{o_f}^f + \sum_{i \in \mathcal{L}_{o_f}^f} \gamma_i^f - \sum_{j \in \mathcal{R}_f} \gamma_j^f \right) + \beta_D^A \left(\sum_{j \in \mathcal{R}_f} \gamma_j^f - \sum_{i \in \mathcal{E}_f} \gamma_i^f \right) \Bigg] +$$

$$\sum_{g \in \mathcal{F}_A} \left[\beta_A^A \left(\gamma_{o_g}^g - \sum_{j \in \mathcal{R}_g} \gamma_j^g \right) + \beta_A^G \left(\sum_{j \in \mathcal{R}_g} \gamma_j^g - \sum_{i \in \mathcal{E}_g} \gamma_i^g \right) \right], \qquad (3.4)$$

where $\gamma_i^f = \sum_{t=\underline{T}_i^f}^{T} z_{it}^f$.

Before digesting this expression, note that it is separated into five terms, corresponding to the costs of the five different types of delay:

$$\begin{pmatrix} \text{departures} \\ \text{at the gate} \\ \text{with engines off} \end{pmatrix} + \begin{pmatrix} \text{departures} \\ \text{leaving the gate} \\ \text{or taxiing} \end{pmatrix} +$$
$$\begin{pmatrix} \text{departures} \\ \text{in the air} \end{pmatrix} + \begin{pmatrix} \text{arrivals} \\ \text{in the air} \end{pmatrix} + \begin{pmatrix} \text{arrivals} \\ \text{taxiing} \end{pmatrix}.$$

Furthermore, γ_i^f is the length of time from the moment f arrives at node i until the end of the time horizon, if f does indeed arrive at i, and 0 otherwise. Then $\gamma_i^f - \gamma_j^f = \sum_{t=\underline{T}_i^f}^{T} z_{it}^f - \sum_{t=\underline{T}_j^f}^{T} z_{jt}^f$ is the amount of time flight f spends getting from i to j, assuming i comes before j, and f reaches both i and j. The function (3.4) is then the desired one, since each flight must reach exactly one runway, one destination, and one immediate successor node of its origin (this last point requires careful construction of the network graph). Finally, note that the term $l_{o_f}^f$ is the time it takes until a departure f properly begins taxiing after removing its blocks, and is assumed to be constant.

3.3.5 The Binary Optimization Problem

The following binary optimization problem then routes flights to achieve the schedule of assigned runways and assigned flight sequences at each runway which were found in P3-1. The model is based on the models of [7] and [9], which were presented to solve the network ATFM problem with and without re-routing, respectively.

P3-2: min $\quad \Phi$

$$\text{s.t.} \quad x^f_{jt} - \left(z^f_{jt} - \sum_{i \in \mathcal{L}^f_j \,:\, t \geq \underline{T}^f_i} z^f_{it} \right) \geq 0, \quad \forall f \in \mathcal{F},\, j \in \mathcal{S}_f,\, t \in \mathcal{T}^f_j, \tag{3.5a}$$

$$\sum_{\substack{f \in \mathcal{F} \,:\, j \in \mathcal{S}_f, \\ t \in \mathcal{T}^f_j}} x^f_{jt} \leq u_{jt}, \quad \forall j \in \mathcal{S} \setminus \mathcal{R},\, t \in \mathcal{T}, \tag{3.5b}$$

$$z^f_{jt} - \sum_{\substack{i \in \mathcal{P}^f_j \,: \\ t - l^f_i \geq \underline{T}^f_i}} z^f_{i,t-l^f_i} \leq 0, \quad \forall f \in \mathcal{F},\, j \in \mathcal{S}_f \setminus \{o_f\},\, t \in \mathcal{T}^f_j, \tag{3.5c}$$

$$z^f_{i\overline{T}^f_i} - \sum_{j \in \mathcal{L}^f_i} z^f_{j\overline{T}^f_j} \leq 0, \quad \forall f \in \mathcal{F},\, i \in \mathcal{S}_f \setminus \mathcal{E}_f, \tag{3.5d}$$

$$\sum_{j \in \mathcal{E}_f} z^f_{j\overline{T}^f_j} = 1, \quad \forall f \in \mathcal{F}, \tag{3.5e}$$

$$\sum_{j \in \mathcal{L}^f_i} z^f_{j\overline{T}^f_j} \leq 1, \quad \forall f \in \mathcal{F},\, i \in \mathcal{S}_f, \tag{3.5f}$$

$$z^f_{j,t-1} - z^f_{jt} \leq 0, \quad \forall f \in \mathcal{F},\, j \in \mathcal{S}_f,\, t \in \mathcal{T}^f_j \setminus \{\underline{T}^f_j\}, \tag{3.5g}$$

$$z^f_{o_f \underline{T}^f_{o_f}} = 1, \quad \forall f \in \mathcal{F}, \tag{3.5h}$$

$$z^f_{r_f, \overline{T}^f_{r_f}} = 1, \quad \forall f \in \mathcal{F}, \tag{3.5i}$$

$$z^f_{r,t+s^r_{\boldsymbol{c}(g),\boldsymbol{c}(f)}} - z^g_{rt} \leq 0, \quad \forall r \in \mathcal{R},\, (g,f) \in \mathcal{H}_r,\, \forall t \in \mathcal{T} \,:\, t \geq \underline{T}^g_r \text{ and }$$
$$t + s^r_{\boldsymbol{c}(g),\boldsymbol{c}(f)} \in \mathcal{T}^f_r, \tag{3.5j}$$

$$z^f_{r',t+s^{(r,r')}_{\boldsymbol{c}(g),\boldsymbol{c}(f)}} - z^g_{rt} \leq 0, \quad \forall (r,r') \in \mathcal{V},\, (g,f) \in \mathcal{H}_{(r,r')},$$
$$\forall t \in \mathcal{T} \,:\, t \geq \underline{T}^g_r \text{ and } t + s^{(r,r')}_{\boldsymbol{c}(g),\boldsymbol{c}(f)} \in \mathcal{T}^f_{r'}, \tag{3.5k}$$

$$z^g_{i,t+s^i_{\boldsymbol{c}(f),\boldsymbol{c}(g)}} - z^f_{it} \leq 0, \quad \forall t \in \mathcal{T}^f_i \text{ s.t. } t + s^i_{\boldsymbol{c}(f),\boldsymbol{c}(g)} \in \mathcal{T}^g_i,$$
$$\forall (f,g) \in \mathcal{W}_i,\, \forall i \in \mathcal{S}, \tag{3.5l}$$

$$z^f_{r_f,t+l^g_{r_g}} - z^g_{r_g,t} \leq 0, \quad \forall t \in \mathcal{T} \text{ s.t. } t \geq \underline{T}^g_{r_g} \text{ and } t + l^g_{r_g} \in \mathcal{T}^f_{r_f},\, \forall (g,f) \in \mathcal{Q}, \tag{3.5m}$$

$$x^f_{rt} = 0, \quad \forall f \in \mathcal{F},\, r \in \mathcal{U}_t,\, t \in \mathcal{T}^f_r, \tag{3.5n}$$

$$z_{it}^f = z_{i\overline{T}_i^f}^f, \quad \forall f \in \mathcal{F},\, i \in \mathcal{S}_f,\, t \in \{\overline{T}_i^f + 1, \ldots, T\}, \tag{3.5o}$$

$$z_{it}^f \in \{0,1\}, \quad \forall f \in \mathcal{F},\, i \in \mathcal{S}_f,\, t \in \{\underline{T}_i^f, \ldots, T\}, \tag{3.5p}$$

$$x_{it}^f \in \{0,1\}, \quad \forall f \in \mathcal{F},\, i \in \mathcal{S}_f,\, t \in \mathcal{T}_i^f. \tag{3.5q}$$

Constraints (3.5a) link the $\boldsymbol{x}$ variables with the $\boldsymbol{z}$ variables, with x_{jt}^f being forced equal to one only if at time t flight f has arrived at node j but not yet at one of its successor nodes. Constraints (3.5b) then limit the number of flights at any node at a given time to the node's capacity, excluding runways (we take care of these in later constraints, using properties of the solution from P3-1).

Constraints (3.5c) state that flight f cannot reach a node j by time t unless it has reached one of its predecessors i by time $t - l_i^f$. Constraints (3.5d) require that a flight f must eventually reach some follower of any node which it reaches, unless that node is its destination, in which case Constraints (3.5e) state that the flight must reach one of its feasible destinations. Constraints (3.5f) state that a flight f can only reach a single successor of any node i (note that the network representation therefore requires careful construction). Constraints (3.5g) enforce monotonicity on the $\boldsymbol{z}$ variables, owing to their definition. Constraints (3.5h) initialize each flight at its origin.

Constraints (3.5i) force a flight to use its assigned runway from P3-1. Constraints (3.5j) state that flights must be processed at each runway in the order determined from P3-1, and be separated by at least the minimum separation time, while Constraints (3.5k) enforce these same ordering and separation requirements for the pairs of flights scheduled on intersecting/closely-spaced parallel runways. Constraints (3.5l) ensure that flights which do not use a common runway (the separation is already incorporated in Phase One for those that use a common runway) are adequately separated at their arrival/departure fix. Note that since arrivals and departures use separate fixes in general, the number of such constraints will be small. All three sets of constraints (3.5j), (3.5k) and (3.5l) are of a much nicer form than usual separation constraints, for two reasons. First, there are only a limited number of pairs of flights for which the constraints need be applied, as determined by the Phase One solution through the sets $\mathcal{H}_r$, $\mathcal{H}_{(r,r')}$ and $\mathcal{W}$. Second, due to the form of the constraints, which state that one set of the "by" variables $\boldsymbol{z}$ must dominate another set by a specified amount. Indeed, in [9] it was shown that such constraints were facet-defining for the polyhedron corresponding to the convex hull of integer solutions to a very similar integer optimization problem.

Constraints (3.5m) ensure that the configuration requirements are respected by ensuring that we process all pairs of flights in the set $\mathcal{Q}$ in the specified order. Note that we have defined the set $\mathcal{Q}$ to be as small as possible while still preventing the operation of illegal configurations, expanding the feasible space of P3-2. Constraints (3.5n) state that a flight may not be processed at a given runway when that runway is not available (for example due to the weather conditions).

Finally, Constraints (3.5o) extend the $\boldsymbol{z}$ variables so that they are constant at every node j beyond the final time at which a flight can feasibly arrive at that node. The reason we need these variables to exist beyond the upper time window is to ensure Constraints (3.5a) correctly define the variables $\boldsymbol{x}$ in the boundary case – if we do not do this, the term in parentheses might be equal to one, even though flight f is not at node j at time t, due to the non-existence of the variable z^f_{it}.

3.3.6 Valid Inequalities

We add the following valid inequalities to strengthen the model P3-2:

$$z^f_{jt} - \sum_{\substack{i\in\mathcal{L}^f_j:\\ |\mathcal{P}^f_i|=1,\,(t+l^f_j)\in\mathcal{T}^f_i}} z^f_{i,t+l^f_j} \geq 0, \quad \forall f\in\mathcal{F},\, j\in\mathcal{S}_f,\, t\in\mathcal{T}^f_j, \tag{3.6a}$$

$$z^f_{j\overline{T}^f_j} - \sum_{\substack{i\in\mathcal{P}^f_j\setminus\mathcal{E}_f:\\ |\mathcal{L}^f_i|=1}} z^f_{i\overline{T}^f_i} \geq 0, \quad \forall f\in\mathcal{F},\, j\in\mathcal{S}_f, \tag{3.6b}$$

$$\sum_{i\in\mathcal{A}_f} z^f_{i\overline{T}^f_i} \leq 1, \quad \forall f\in\mathcal{F}, \forall\, antichains\ \mathcal{A}_f\subset\mathcal{S}_f. \tag{3.6c}$$

Inequalities (3.6a) and (3.6b) are *fork* and *joint* inequalities, respectively. A fork is a node with a single possible predecessor node, and a joint is one with a single possible following node. The fork inequalities exploit the fact that at a fork, to get to any of its following nodes, a flight must have first arrived at the fork node itself. The joint inequalities exploit the fact that if a flight gets to any predecessor of the joint node, then it must get to the joint node (unless the predecessor is the flight's final destination). Inequalities (3.6c) state that if there is a set $\mathcal{A} \subset \mathcal{S}_f$ of nodes for which, based on the structure of the network, we must have $\sum_{j \in \mathcal{A}} \mathbb{I}_{\{\text{flight } f \text{ reaches } j\}} \leq 1$, then only one of the corresponding variables $z^f_{j\overline{T}^f_j}$, $j \in \mathcal{A}$, can be nonzero. All Inequalities (3.6a) – (3.6c) were introduced in [7], and we direct the reader to that paper for further details.

3.4 Computational Experience

In this section we present extensive computational experience which seeks to answer several key questions regarding the effectiveness of the solution approach we have presented, in particular:

- Are our key assumptions valid?
- Is the methodology computationally tractable?
- Would the use of the methodology result in significant benefits in practice?

In order to answer these questions, we focus on two international airports: BOS and DFW. We utilize data from historic days of operation at these airports, 11/02/2009 at DFW and 9/28/2010 at BOS. In particular, we make use of the following data sources:

- Hourly METAR (see [33]) weather forecasts, from which runway availability is determined;
- Airport Surface Detection Equipment, Model X (ASDE-X) data flight track data, indicating individual flight positions and timing, velocity, acceleration, and heading, amongst other fields;

- ASPM OOOI data - the “on/off/out/in” times for all scheduled flights, indicating wheels-on, wheels-off, gate-out and gate-in times.

All experiments were performed using the software package GUROBI 5.0 on a computer with an Intel®Core$^{\text{TM}}$ i7-860 Processor (8MB Cache, 2.80GHz) and 16GB DDR3 RAM, running Ubuntu Linux 10.04. The solver time limit was set to 1200s for each of P3-1 and P3-2.

For each historic time period considered, we input the historic weather, flight schedules, etc, as data to our model. In all of the tables which follow, our time horizon incorporates one hour's worth of arrivals and departures, and we use a discretization of 20 seconds per time interval. The values which most parameters take should be self-explanatory, but we note that we set $(\beta_G, \beta_A^A, \beta_D^A, \beta_A^G, \beta_D^G) := (0.5, 1.2, 1.3, 1.0, 1.1)$, as well as the following:

$$\underline{T}_{o_f}^f \quad := \quad \begin{cases} \text{actual pushback time of flight } f, & \text{if } f \text{ a departure,} \\ \text{actual touchdown time of flight } f \text{ less a constant,} & \text{if } f \text{ an arrival;} \end{cases}$$

$$\mathcal{U}_t \quad := \quad \text{the set of runways which cannot be used at time } t\text{, based on the historic wind conditions.}$$

3.4.1 Model Validation and Computational Tractability

Tables 3.3 and 3.4 present resulting computation times and solution values for several historic time periods. The purpose of these two tables is i) to demonstrate the suitability of our two-phase approach, and ii) to demonstrate computational tractability on real-world instances.

The first thing to note is that the optimal values from Phase One and two differ only very slightly – aside from the greatest difference of 2.1%, the others are at most 1.1%, and in the former case 1% of the gap could be explained by the sub-optimality of P3-2. This indicates that our two-phase approach results in solutions which are very close to optimal. This supports our fundamental belief that the runways represent the key bottleneck of the system and justifies our particular two-phase approach.

The second observation we make is that the computation times are low across all

Flights	**Objective Values**			**% Opt. Gap**		**Comp. Times (s)**		
	P3-1	**P3-2**	**% Diff.**	**P3-1**	**P3-2**	**P3-1**	**P3-2**	**Total**
155	12,806.2	13,073.6	2.1	0.0	1.0	120	1286	1430
175	14,365.3	14,530.4	1.1	0.0	0.0	372	1071	1465
153	12,513.7	12,586.7	0.6	0.0	0.0	64	129	211
155	12,879.2	12,972.0	0.7	0.0	0.0	75	284	379
168	13,988.0	14,064.8	0.5	0.0	0.0	340	187	546
171	14,584.6	14,683.0	0.7	0.0	0.0	299	284	600
159	13,018.4	13,140.9	0.9	0.0	0.0	252	533	806
153	12,483.2	12,556.2	0.6	0.0	0.0	241	205	463

Table 3.3: Computational tractability and a bound on the optimality gap, using data from 11/2/2009 at DFW. The objective value for P3-1 above has had the component due to the configuration change penalty removed.

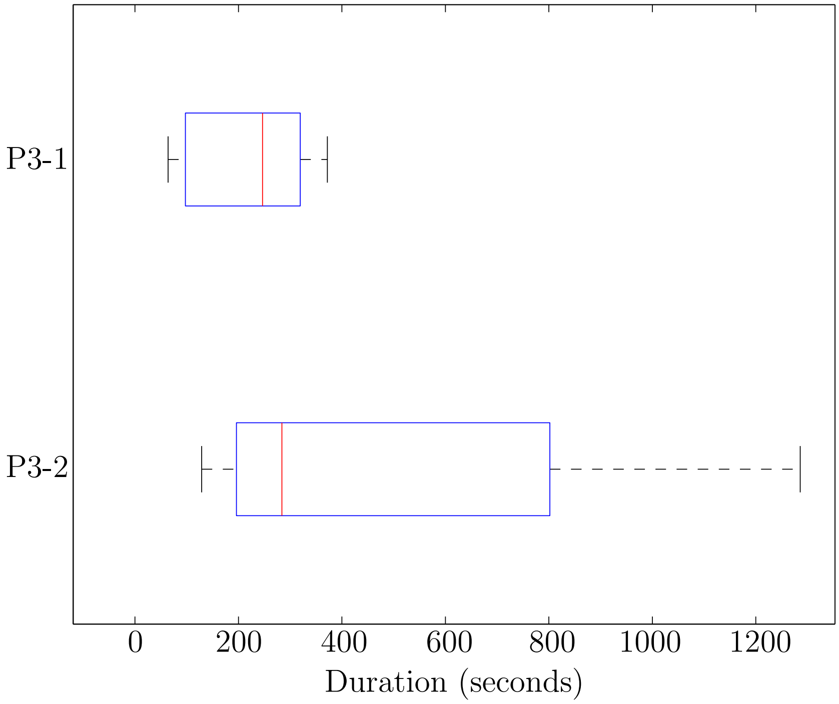

Figure 3-2: Boxplot of P3-1 and P3-2 computation times for DFW.

Flights	Objective Values			% Opt. Gap		Comp. Times (s)		
	P3-1	P3-2	% Diff.	P3-1	P3-2	P3-1	P3-2	Total
90	4,687.6	4,696.3	0.2	0.0	0.0	252	147	418
91	4,619.8	4,639.3	0.4	0.0	0.0	233	142	388
80	5,114.7	5,117.7	0.1	0.0	0.0	143	16	168
63	3,111.2	3,128.7	0.6	0.0	0.0	93	52	161
63	3,168.2	3,179.2	0.3	0.0	0.0	198	15	229
71	3,877.1	3,928.3	1.3	0.0	0.0	246	1200	1457
59	2,681.4	2,695.0	0.5	0.0	0.0	255	59	325
63	3,512.9	3,524.4	0.3	0.0	0.0	161	17	187

Table 3.4: Computational tractability and a bound on the optimality gap, using data from 9/28/2010 at BOS. The objective value for P3-1 above has had the component due to the configuration change penalty removed.

instances, which can also be seen in Figures 3-2 and 3-3. These figures highlight the consistency of the P3-1 computations times, while also showing that in some instances, P3-2 can take significantly longer than the median. In these cases, however, it is true that we typically have a good solution much earlier than termination.

3.4.2 Benefits Assessment

We have demonstrated above that our approach leads to solutions which are very close to optimal in a practical amount of time. Now we aim to assess the potential benefit that can be gained in practice from using the methodology. In Tables 3.5 and 3.6 we present statistics both for what actually occurred on the historic days of operation considered and for our optimized schedule. In particular, we compare the mean and standard deviation of the times taken for flights to traverse part of the system – for arrivals, we record the time from touch-down until arrival at the gate, and for departures we record the time from pushback until take-off. Ideally, we would present the overall system traversal times, from fix to gate or from gate to fix, but due to lack of historical fix-at times we could not make a comparison of these times. Nevertheless, the results presented give a good indication of the model's benefits. Indeed, since the objective function weights $\boldsymbol{\beta}$ place a higher emphasis on reducing

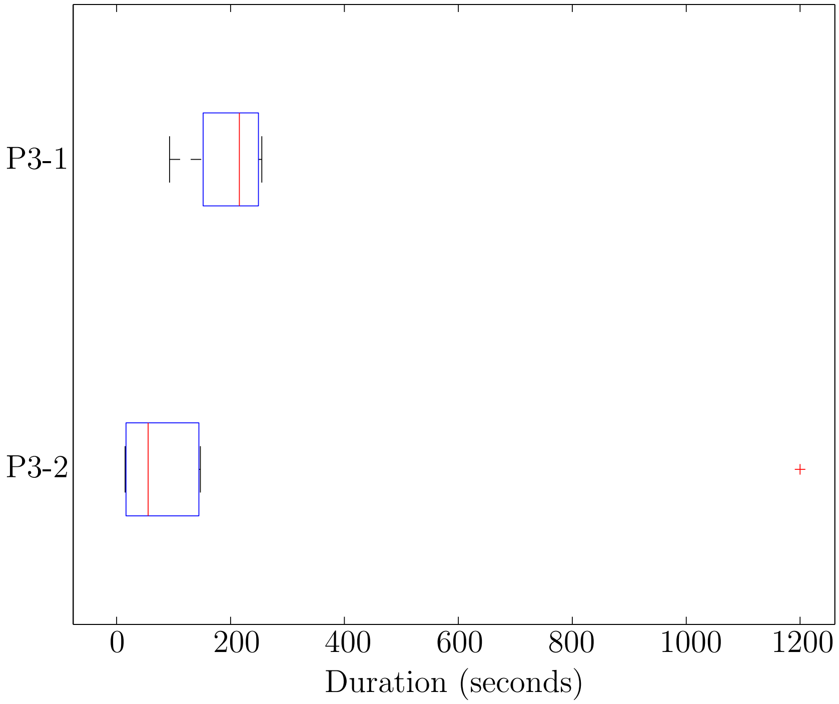

Figure 3-3: Boxplot of P3-1 and P3-2 computation times for BOS.

airborne delays (as is appropriate), it is fair to say that the ensuing benefits assessment is conservative, since it compares the less-prioritized surface traversal times. Figures 3-4 – 3-7 present boxplots of these statistics, separated into arrival and departure groups.

Overall, we can see that in all cases the average optimized ground times are lower than the historic ones, with reductions of 5-14% at DFW and 7-25% at BOS. As mentioned above, the reductions to air delays could be expected to be at least as good as this. For arrivals, however, surface traversal times are in general worse in the optimized solution, due to the relatively low weight placed on arrival taxi times – the model sacrifices these slightly for the sake of reduced air delays and departure taxi times. We also observe that the spread of the times is reduced in almost all instances, meaning that different flights are treated more equally. Finally, we note that there is indeed a non-negligible element of gate-holding of departures, which appears to be positively correlated with the number of flights (and hence congestion), as would be expected.

	Optimized Surface Times				Historic Surface Times			Comp.
$\lvert\mathcal{F}\rvert$	Dep. G.H.	Dep.	Arr.	Avg.	Dep.	Arr.	Avg.	Time (s)
155	1.8/2.7	9.5/1.8	10.8/2.6	10.2/2.3	13.0/3.9	8.9/3.9	10.7/4.4	1430
175	2.2/3.2	9.2/1.9	10.7/2.4	9.9/2.2	12.8/3.2	9.3/4.4	11.2/4.2	1465
153	1.0/1.8	9.5/2.0	10.6/2.3	10.0/2.2	13.5/3.0	9.2/4.9	11.6/4.5	211
155	1.0/1.2	9.6/1.9	11.1/2.4	10.4/2.3	13.5/3.9	9.0/3.7	11.0/4.4	379
168	0.9/1.4	9.3/1.9	11.8/2.8	10.5/2.7	13.0/2.9	10.1/5.7	11.6/4.7	546
171	2.1/3.0	10.0/2.1	11.4/2.5	10.7/2.4	13.6/3.2	8.9/3.9	11.3/4.3	600
159	1.3/1.6	9.1/1.8	10.8/2.4	10.0/2.3	13.9/3.1	9.1/3.9	11.4/4.3	806
153	0.7/1.0	9.0/1.8	11.1/2.1	10.1/2.2	13.4/3.8	9.6/3.6	11.2/4.1	463

Table 3.5: Comparison of optimized and historic surface times, using data from 11/2/2009 at DFW. Statistics given are the relevant mean and standard deviation, in minutes per flight for gate-holding and from pushback to wheels-off for departures, and from wheels-on to gate-in for arrivals.

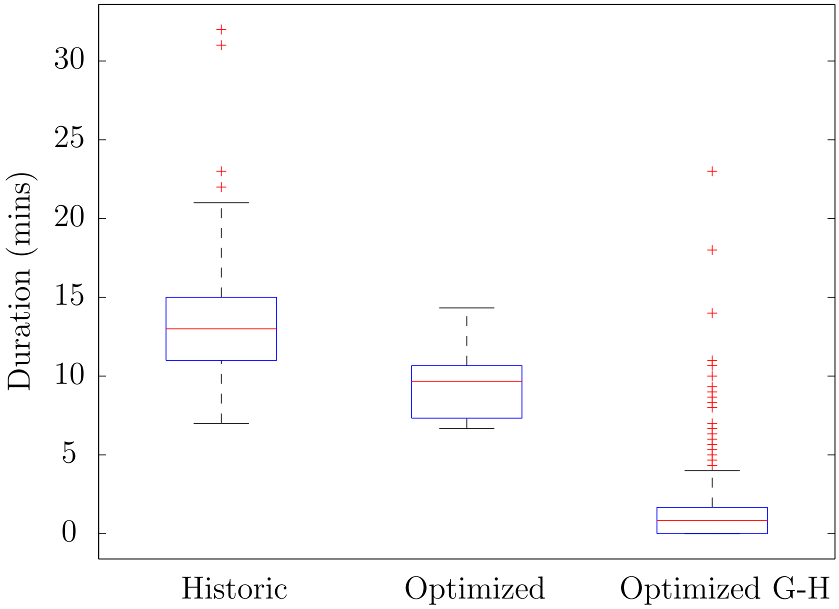

Figure 3-4: Boxplot of historic and optimized surface times for departures at DFW, as well as optimized gate-holding.

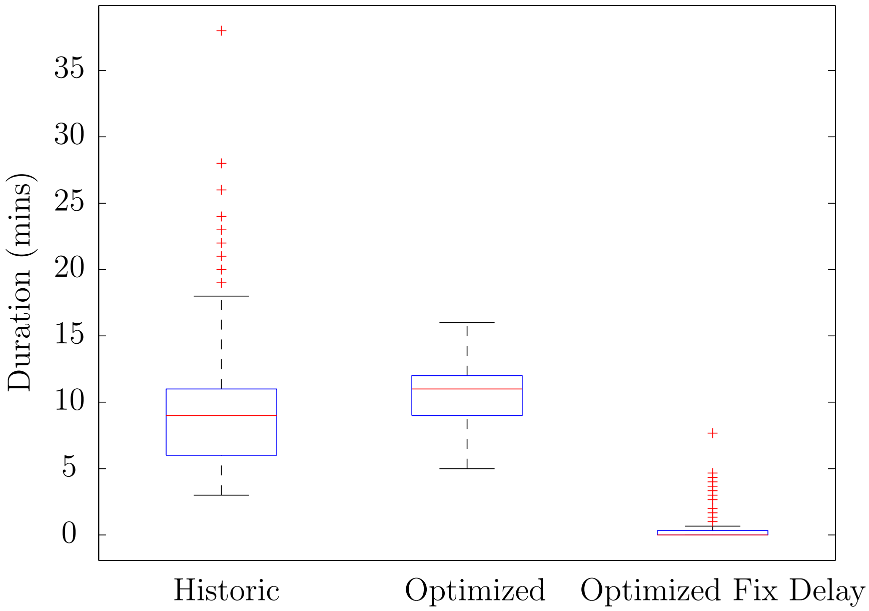

Figure 3-5: Boxplot of historic and optimized surface times for arrivals at DFW, as well as optimized fix-delays (i.e. speed control before reaching fix).

	Optimized Surface Times				**Historic Surface Times**			**Comp.**
$\lvert\mathcal{F}\rvert$	**Dep. G.H.**	**Dep.**	**Arr.**	**Avg.**	**Dep.**	**Arr.**	**Avg.**	**Time (s)**
90	1.6/2.0	13.7/0.8	4.4/1.2	9.3/4.8	18.7/6.0	5.2/1.1	12.4/8.1	418
91	0.8/1.0	14.0/0.8	4.5/1.2	9.6/4.8	16.7/4.8	6.1/1.7	11.6/6.5	388
80	2.2/3.9	16.4/0.7	5.0/1.1	12.2/5.6	17.6/5.3	5.9/2.0	13.1/7.2	168
63	0.5/0.7	16.6/0.6	4.9/1.1	10.3/5.9	18.2/8.4	6.5/1.7	12.2/8.4	161
63	0.5/0.7	16.0/0.8	5.0/1.1	10.1/5.6	18.5/8.2	4.4/1.1	10.9/9.0	229
71	1.3/1.5	11.6/1.0	8.2/0.8	10.1/1.9	16.6/4.4	5.1/0.9	11.4/6.6	1457
59	2.6/5.2	13.8/0.7	4.6/1.3	8.5/4.7	14.7/4.9	5.8/1.3	9.9/5.7	325
63	0.9/1.1	16.2/0.9	4.9/1.1	10.5/5.7	17.1/6.9	6.6/1.8	11.3/7.1	187

Table 3.6: Comparison of optimized and historic surface times, using data from 9/28/2010 at BOS. Statistics given are the relevant mean and standard deviation, in minutes per flight for gate-holding and from pushback to wheels-off for departures, and from wheels-on to gate-in for arrivals.

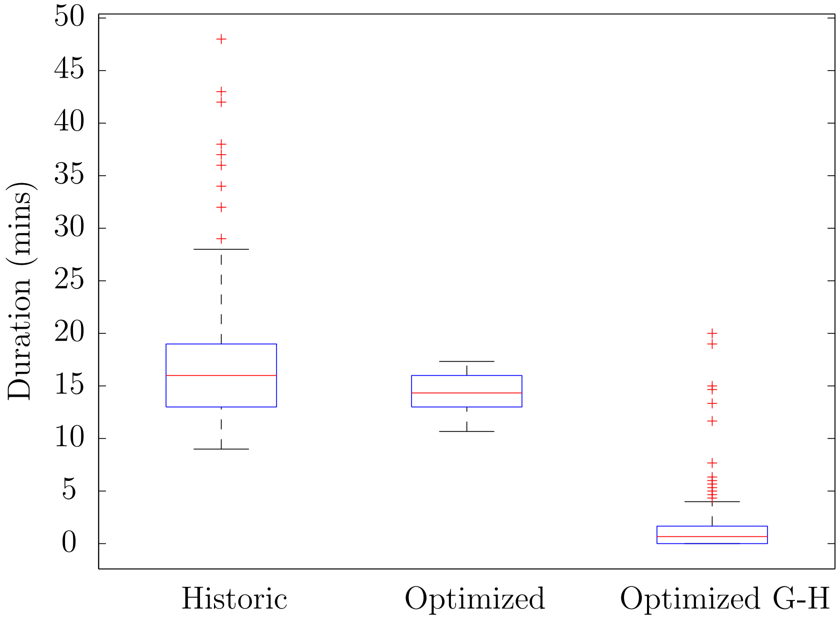

Figure 3-6: Boxplot of historic and optimized surface times for departures at BOS, as well as optimized gate-holding.

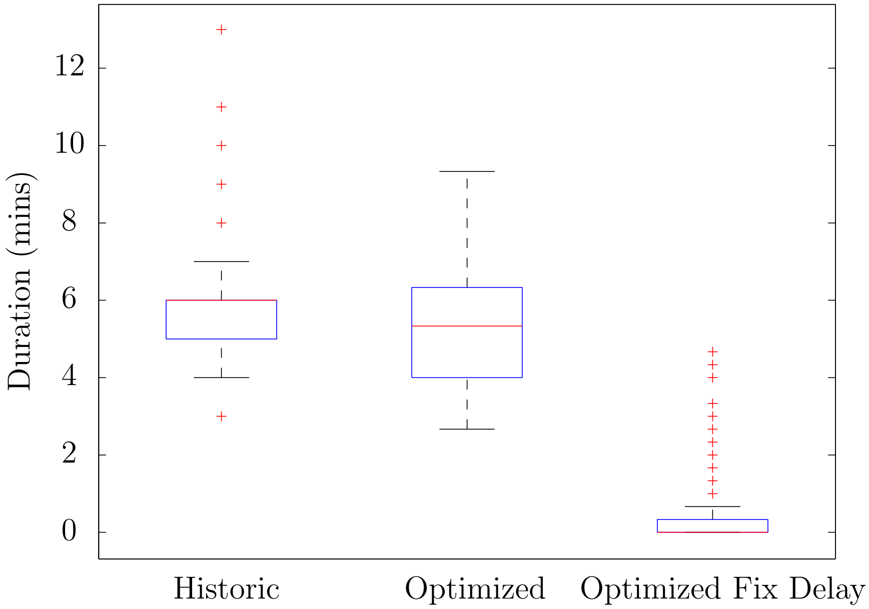

Figure 3-7: Boxplot of historic and optimized surface times for arrivals at BOS, as well as optimized fix-delays (i.e. speed control before reaching fix).

Figures 3-8 and 3-9 present the above results in terms of their impact on surface congestion, comparing historical surface congestion with that of the optimized solution, for a typical optimization time period. Note that any departure which has pushed back and not taken off, and any arrival which has landed but not reached the gate contributes to these tallies. Also note that any shift in the arrival congestion between the historic and optimized data reflects that exact fix-at times were not available as inputs to our optimization, and so the release times of arrivals into the system were approximated based on their historic touchdown times.

From these figures we can see that the congestion due to arrivals is not significantly changed due to the optimization. However, it is clear that the impact on congestion due to departures is very significant, especially in the case of BOS. This is due to two phenomena present in the optimal policies – first, almost any delays to departures are incurred at the gate with engines off, and second, aircraft do not queue up at runways, but rather leave the gate at just the right time to reach their assigned runway for take-off. This phenomenon is desirable in an ideal world where the availability of flights for processing is deterministically known, however, the incorporation of robustness to delays in aircraft availability, which we shall outline in Chapter 4, would be expected to add a small buffer to these optimized taxi times.

We now return to summarize our answers to the questions introduced at the beginning of this section, using the above computational experience at DFW and BOS.

- Our fundamental assumption about the nature of airport capacity is a reasonable one, as demonstrated by the small differences observed between the values of the first and second phases of optimization.

- The computational tractability of the approach is promising for possible implementation in the future, with the complete optimization typically taking 5-10 minutes on a desktop computer, and always less than half an hour.

- The methodology leads to significant reductions in delays from the levels historically observed. This is clearly a great benefit in itself, but also results in

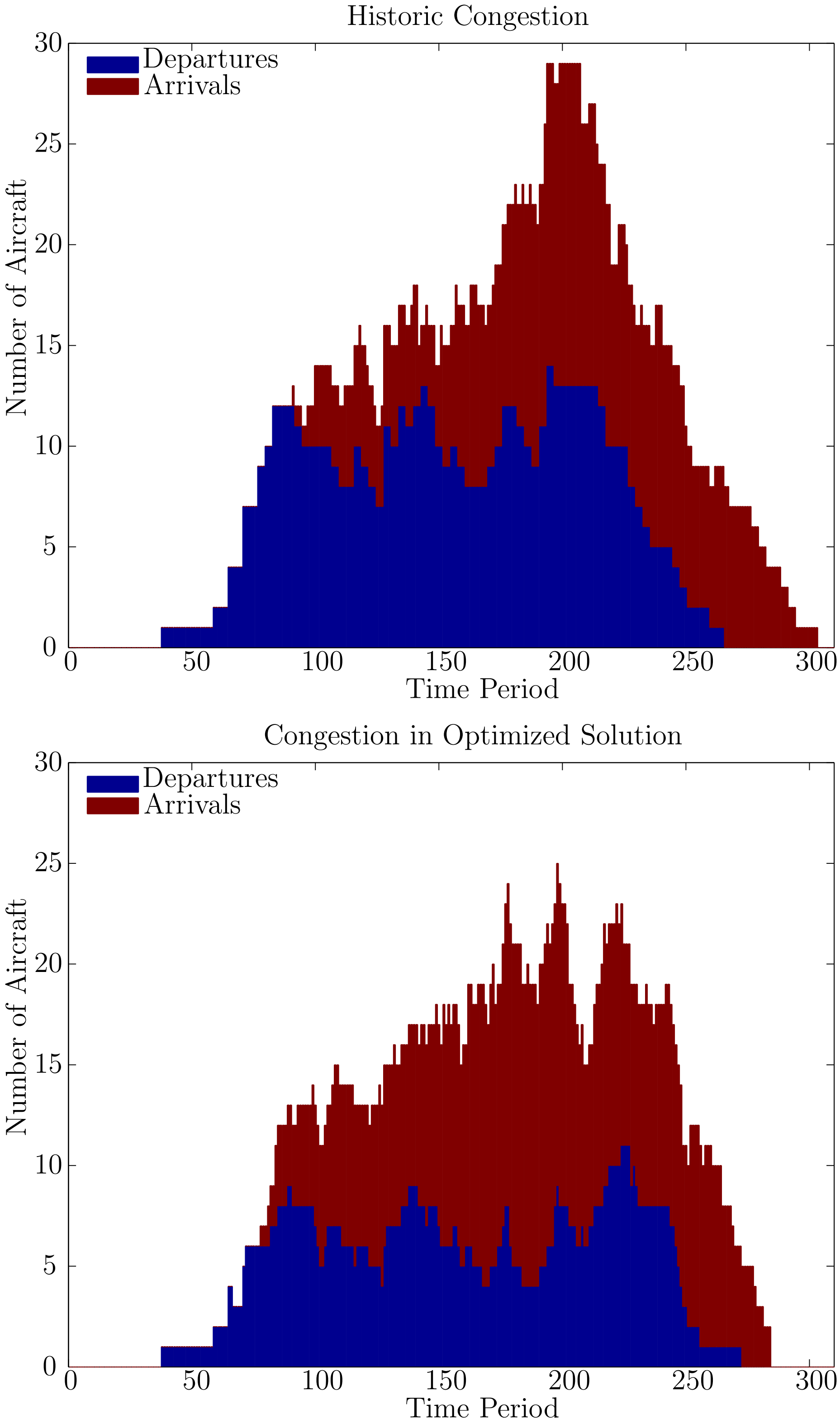

Figure 3-8: Contribution of scheduled flights in the optimization time window to surface congestion at DFW.

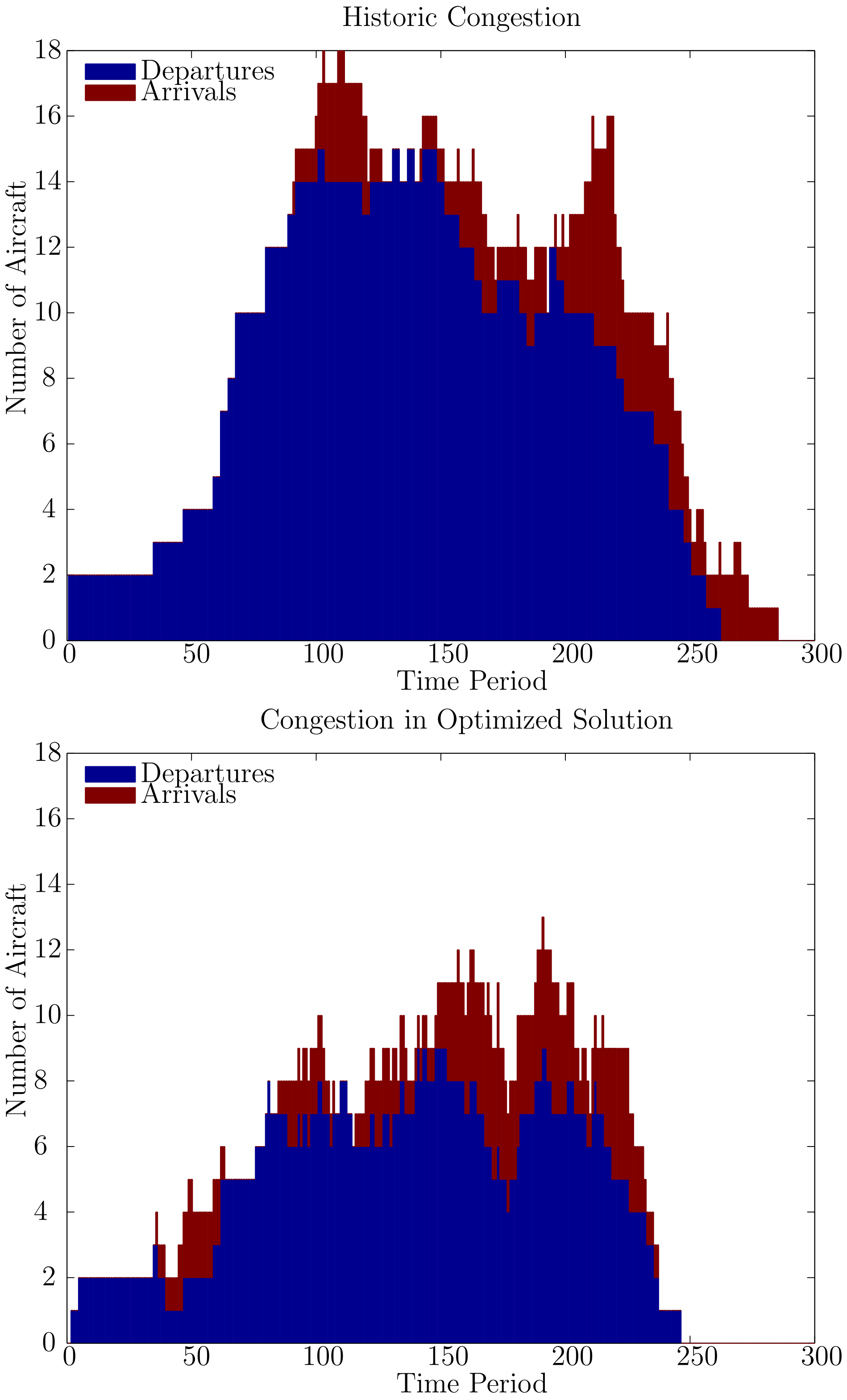

Figure 3-9: Contribution of scheduled flights in the optimization time window to surface congestion at BOS.

increased throughput, less congestion of the airport surface and near-terminal airspace, less fuel burn and hence reduced fuel costs and associated emissions.

3.5 Summary

In this chapter, we have presented a novel, integrated approach to solving the entirety of key air traffic flow management problems faced at an airport. Through computational experiments using historic data from BOS and DFW airports, we have shown the methodology to be both tractable (in a practical sense) and of significant potential benefit. The models have the potential to influence ATFM on a very broad scale when one considers the optimization of a nationwide or supranational airspace as a combination of optimizations of through-airport flows, the airports being where many important and difficult decisions need to be made.

Chapter 4

Optimization of Air Traffic at Airports in the Presence of Uncertainty

In this chapter, we address the implications of the real world's dynamic and uncertain nature on the implementation of the methodology presented in Chapter 3. In particular, we propose modifications to the model which consider uncertainty, one of the key aims being to reduce the negative impact of randomness on the optimized solutions, resulting in greater reliability of these solutions in practice.

4.1 Introduction

In Chapter 3, we assumed all data inputs to the model P3-1 to be deterministically known, when indeed there are many sources of uncertainty in the AOOP, the most notable of which are

a) the times at which flights are released into the system (i.e., the earliest possible pushback times of departures from their gates, and the earliest possible times of arrivals at their arrival fixes),

b) the travel times of flights in the network, and

c) the times at which different runways are available for use.

Uncertainty of types (a) and (b) is significant because in an optimal solution to the deterministic AOOP we process flights at runways based on the time window during which we assume them to be available for processing. In reality, however, flights are often delayed and may not be able to reach their runway at the optimized time. This could be due, for example, to delayed aircraft, crew, or passengers, bottlenecks at gates, or variations in taxi speed between airlines and/or pilots.

Uncertainty of type (c) clearly also has the ability to influence our problem to a great extent – what happens if we assign flights to use a certain runway and it turns out that runway is not available due to unfavorable winds?

While there has been much work on the various airport optimization subproblems in the literature, there is a dearth of work considering stochastic conditions. Notably, [41] recently proposed a two-stage stochastic optimization approach for the runway sequencing problem under uncertainty. The same study, as well as [26], assessed the impact of uncertainty when implementing runway optimization policies. Also, [32] considered the optimization of near-terminal air routes under uncertainty in weather conditions.

After determining the key sources of uncertainty, the next step in developing an approach to solve the AOOP which takes uncertainty into account is to consider how such a methodology will be implemented. For example, when will the optimization be solved, and re-solved? When will uncertainty be resolved and data updated? How far in advance must we have a flight's schedule fixed? What happens when the optimized solution is infeasible in practice?

We envision implementing our approach on a rolling horizon basis with regular re-solves. We consider half-hourly re-solves to be reasonable given that our runway time horizon is one hour, and the computational experience of Chapter 3. As mentioned in Chapter 3, the phase one optimization problem P3-1 produces a flight type sequence, and not an explicit sequence of flights. Indeed, this aspect of P3-1 leads us to an implementation strategy which is more robust to uncertainties in the earliest possible touchdown/takeoff times of flights: as pointed out in [41], there is less uncertainty in the flight type mix we will have available for runway processing at any moment in the future, than in the specific flights that will be available. In this way, we can wait until we know the actual availability of flights (i.e., when uncertainty is revealed), before solving an assignment problem to assign flights to the flight type slots from P3-1. Indeed, [41] capitalized on this phenomenon in their proposed stochastic runway scheduling approach.

While that paper focused on runway scheduling only, we here again consider the full scope of the AOOP, notably including runway sequencing and flight routing. Furthermore, while their stochastic optimization approach suffered from the curse of dimensionality, necessitating approximate sampling-based approaches, we shall here draw on the techniques of robust optimization when we consider uncertainty in flights' earliest possible touchdown/takeoff times. In this way, we do not need to know the exact probability distributions of the uncertainties involved, resulting in increased tractability properties, and also eliminating the need to assign such distributions artificially to these random quantities. Robust optimization draws on the laws of large numbers – while individual uncertainties may be unpredictable, as the number of random variables grows large, uncertainties tend to cancel out, and we can predict averages fairly well. We refer the reader to [8] and [4] for an introduction to robust optimization.

As will be described in Section 4.3, we are nevertheless able to make use of stochastic optimization when incorporating uncertainty in runway availability, ensuring tractability by modeling in such a way that problem size remains of the same order of magnitude. We refer the reader to [39] for an introduction to stochastic optimization.

Outline of Chapter

The remainder of this chapter is outlined as follows:

In Section 4.2, we first propose a robust optimization approach to solving the AOOP which considers the first two types of uncertainty, (a) and (b), mentioned above. We then demonstrate how to compute uncertainty sets, needed for the optimization, from historic data. Finally, we present computational experience and derive insights regarding both the effectiveness of the robust approach, and the impact of uncertainty on the deterministic approach.

In Section 4.3, we present an extension to the robust approach, which uses stochastic optimization to consider uncertainty in runway availability. We also present computational results demonstrating its potential impact.

This chapter builds on the Phase One approach of Chapter 3, and the corresponding assumption that flights can travel along shortest paths to their assigned runways will be in effect. Of course, as in Chapter 3, this assumption is relaxed in Phase Two of our approach.

4.2 Incorporating Uncertainty in Flights' Earliest Possible Touchdown/Takeoff Times

We begin our analysis by considering uncertainty in the release times of flights into our system, and also in flights' travel times (i.e., uncertainty of types (a) and (b) above), which affects the optimization problem P3-1 through the parameters $\underline{T}_r^f$. Below we describe precisely what these (as well as related random variables) mean in the context of this chapter:

$\underline{T}_r^f$ = the value of flight f's "earliest possible touchdown/takeoff time" at runway r, as used in P3-1. We shall refer to this as its *nominal value*, and we take it to be its scheduled value;

$\tilde{\underline{T}}_r^f$ = the random variable representing flight f's earliest possible touchdown/takeoff time at runway r in practice. Typically, its realized value will not be known when we solve our optimization problem, however, we shall build a model under which we know an *uncertainty set* to which it belongs.

The nominal values $\underline{T}_r^f$ appear in the right-hand set of inequalities of Constraints

(3.2i), which is duplicated below for convenience:

$$\sum_{t=1}^{\tau} \psi^{\boldsymbol{i}}_{rt} \le \sum_{f\in\mathcal{F}_{\boldsymbol{i}}:\tau\ge\underline{T}^f_r,\, r\in\mathcal{R}_f} \varphi^f_r, \quad \forall \boldsymbol{i}\in\mathcal{C},\, r\in\mathcal{R}_{\boldsymbol{i}},\, \tau\in\mathcal{T}. \tag{4.1}$$

Recall these constraints state that the number of flights of type $\boldsymbol{i}$ which are assigned to runway r to take off or land by time t can be at most the number of flights of type $\boldsymbol{i}$ which are assigned to runway r and could have (according to the nominal values $\underline{T}^f_r$) taken off/landed there by time t.

Then the infeasibility in practice of a direct implementation of feasible solutions to P3-1 can only be caused when the realized values of $\underline{\tilde{T}}^f_r$ are greater than their corresponding nominal values. Indeed, if the realization of even one of the random variables $\underline{\tilde{T}}^f_r$ turns out to be greater than the corresponding nominal value by any amount, then a solution to P3-1 may be infeasible in practice. If we wish to protect against variations in $\underline{\tilde{T}}^f_r$ across the uncertainty set to which it belongs, so that we can be sure of the feasibility of our solution in practice, then it is clear that using its largest possible realized value in our deterministic optimization problem would achieve such a result. However, the problem with this approach is that it would be too conservative, resulting in a solution in which many flights would be withheld from processing for too long.

4.2.1 A Robust Optimization Approach

We now present a robust integer optimization model RP4-1, based on P3-1, which considers uncertainty in flights' earliest possible touchdown/takeoff times. First, we define the following additional decision variables:

$\Gamma^{\boldsymbol{i}}_{rt} =$ the "protection level" against delays in the earliest possible touchdown/
takeoff times of flights of type $\boldsymbol{i}$ assigned to runway r by time t.

Here, we can interpret protection level $\Gamma^{\boldsymbol{i}}_{rt}$ as a measure of the maximum deviation from the schedule (i.e., from the nominal values used in our optimization problem) against which we protect, for flights of type $\boldsymbol{i}$ which are assigned to runway r and could have taken off/landed by time t.

Our model is then

$$\begin{aligned}
\textbf{RP4-1:}\ \min \quad & \Psi - C \cdot \sum_{\boldsymbol{i}\in\mathcal{C}} \sum_{r\in\mathcal{R}_{\boldsymbol{i}}} \sum_{t\in\mathcal{T}} \Gamma^{\boldsymbol{i}}_{rt} \\
\text{s.t.} \quad & (3.2\text{a}) - (3.2\text{h}), \\
& \sum_{\substack{f\in\mathcal{F}_{\boldsymbol{i}}\,:\,r\in\mathcal{R}_f,\\ t\geq \overline{T}^f_r - l^r_{\boldsymbol{i}}+1}} \varphi^f_r \leq \sum_{\tau=1}^{t} \psi^{\boldsymbol{i}}_{r\tau} \leq \sum_{\substack{f\in\mathcal{F}_{\boldsymbol{i}}\,:\,r\in\mathcal{R}_f,\\ t\geq \underline{T}^f_r}} \varphi^f_r - \Gamma^{\boldsymbol{i}}_{rt},\ \forall \boldsymbol{i}\in\mathcal{C},\ r\in\mathcal{R}_{\boldsymbol{i}},\ t\in\mathcal{T}, \\
& \hfill (4.2\text{a}) \\
& \Gamma^{\boldsymbol{i}}_{rt} \leq N^{\alpha}_{\boldsymbol{it}}, \quad \forall \boldsymbol{i}\in\mathcal{C},\ r\in\mathcal{R}_{\boldsymbol{i}},\ t\in\mathcal{T}, \qquad (4.2\text{b}) \\
& (3.2\text{j}) - (3.2\text{n}).
\end{aligned}$$

Note that Ψ is as in P3-1, C is a sufficiently large constant, and α and each $N^{\alpha}_{\boldsymbol{it}}$ are nonnegative constants, the details of which will be outlined below in Section 4.2.2.

The left-hand inequalities of Constraints (4.2a) are identical to those of Constraints (3.2i) in P3-1, and the right-hand inequalities state that the maximum number of flights of type $\boldsymbol{i}$ that can take off/land at runway r by time t is the number of flights of type $\boldsymbol{i}$ assigned to runway r which could have taken off/landed there by time t based on the nominal values $\underline{T}^f_r$, *less our protection level.* Hence, given that $\boldsymbol{\Gamma} \geq \mathbf{0}$ (based on C being sufficiently large), fewer flight type slots will be assigned by time t in an optimal solution of this optimization problem than in one of P3-1, i.e., we are holding some flights back from processing, in anticipation of delays.

Constraints (4.2b) restrict the protection levels to a maximum level of conservativeness. Exactly how this level is chosen will become more clear in the remainder of this section.

Before tailoring the constants $N^{\alpha}_{\boldsymbol{it}}$ of RP4-1, we shall first present a model of uncertainty in Assumption 4.1, and then prove a corresponding result in Proposition 4.1 which will help us to understand our robust approach.

Assumption 4.1. *For every flight type $\boldsymbol{i}$, and every time t, at most $N^{\alpha}_{\boldsymbol{i}t}$ flights of type $\boldsymbol{i}$ may be delayed at time t. Here, we say that flight f is delayed at time t if $\underline{\tilde{T}}^f_r > t \geq \underline{T}^f_r$, for some r.*

Proposition 4.1. *An optimal solution to RP4-1 will always be feasible in practice, for any realized uncertainty coming from the assumed model.*

Proof. Let $(\boldsymbol{\omega}^{*\prime}, \boldsymbol{\varphi}^{*\prime}, \boldsymbol{\psi}^{*\prime}, \boldsymbol{\chi}^{*\prime}, \boldsymbol{\Gamma}^{*\prime})'$ be an optimal solution to RP4-1. Recalling that we are not strictly enforcing flight deadlines, we first note that having feasibility of this solution in practice, under the assumed model of uncertainty, is equivalent to having

$$\sum_{\tau=1}^{t} \psi^{*\boldsymbol{i}}_{r\tau} \leq \sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \underline{\tilde{T}}^f_r} \varphi^{*f}_r, \quad \forall \boldsymbol{i} \in \mathcal{C},\, r \in \mathcal{R}_{\boldsymbol{i}},\, t \in \mathcal{T}.$$

In other words, there is a mapping of flights to flight type slots that uses the optimized flight-runway assignments and is feasible in practice, in the sense that each flight can make its assigned slot.

Suppose now that there is a realization of uncertainty such that for some $\boldsymbol{i} \in \mathcal{C}$, $r \in \mathcal{R}_{\boldsymbol{i}}$, $t \in \mathcal{T}$

$$\sum_{\tau=1}^{t} \psi^{*\boldsymbol{i}}_{r\tau} > \sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \underline{\tilde{T}}^f_r} \varphi^{*f}_r.$$

Then

$$\sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \underline{\tilde{T}}^f_r} \varphi^{*f}_r < \sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \underline{T}^f_r} \varphi^{*f}_r - \Gamma^{*\boldsymbol{i}}_{rt}$$

by feasibility in RP4-1. We have one of two cases:

$$\text{Case i) } \Gamma^{*\boldsymbol{i}}_{rt} = N^{\alpha}_{\boldsymbol{i}t}; \text{ or Case ii) } \Gamma^{*\boldsymbol{i}}_{rt} < N^{\alpha}_{\boldsymbol{i}t}.$$

We will show a contradiction – that for each of these cases, in fact

$$\Gamma^{*\boldsymbol{i}}_{rt} \geq \sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \underline{T}^f_r} \varphi^{*f}_r - \sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \underline{\tilde{T}}^f_r} \varphi^{*f}_r.$$

Case i)

By our assumption on the nature of the uncertainty, $\tilde{\underline{T}}_r^f > t \geq \underline{T}_r^f$ at most $N_{\boldsymbol{it}}^{\alpha}$ times for flights of type $\boldsymbol{i}$ at time t, and so at most $N_{\boldsymbol{it}}^{\alpha}$ terms in the summation $\sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \underline{T}_r^f} \varphi_r^{*f}$ can be greater than the corresponding terms in the summation $\sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \tilde{\underline{T}}_r^f} \varphi_r^{*f}$.

Case ii)

Since our optimal solution is feasible, it clearly satisfies

$$\sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \underline{T}_r^f} \varphi_r^{*f} - \Gamma_{rt}^{*\boldsymbol{i}} \geq \sum_{\tau=1}^{t} \psi_{r\tau}^{*\boldsymbol{i}} \geq 0.$$

Then, since C is large, we must have

$$\Gamma_{rt}^{*\boldsymbol{i}} = \sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \underline{T}_r^f} \varphi_r^{*f},$$

or else we would have Case i). It then follows that

$$\Gamma_{rt}^{*\boldsymbol{i}} \geq \sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \underline{T}_r^f} \varphi_r^{*f} - \sum_{f \in \mathcal{F}_{\boldsymbol{i}} \,:\, r \in \mathcal{R}_f,\, t \geq \tilde{\underline{T}}_r^f} \varphi_r^{*f}.$$

Hence any optimal solution to the robust problem is feasible in practice under the assumed model of uncertainty. □

4.2.2 Modeling Uncertainty to Determine the Robust Optimization Parameters

We are now interested in computing the right-hand sides $N_{\boldsymbol{it}}^{\alpha}$ of Constraints (4.2b) of our robust model RP4-1. In other words, we shall determine how to set the degree of robustness of the model appropriately, based on i) historic data, and ii) our desired degree of conservativeness.

For the remainder of this section, we shall focus on uncertainty of type (a), ignoring uncertainty of type (b). A similar analysis could be performed for the latter type of uncertainty in order to incorporate it into the computation of the values of $N^{\alpha}_{\boldsymbol{i}t}$, and hence into the model.

Consider the possibility of any flight being available for release into the system at time t. Let us define the following:

$\underline{T}^f$ = the value of flight f's earliest possible release time into the system (i.e., its earliest possible pushback time from its gate, or its earliest possible time at its arrival fix) used in P3-1. We shall refer to this as its *nominal value*, and we take it to be its scheduled value;

$\tilde{\underline{T}}^f$ = the random variable representing flight f's actual earliest possible release time into the system in practice;

Z^f = $\tilde{\underline{T}}^f - \underline{T}^f$, a random variable.

Note that because we are ignoring uncertainty of type (b), $\tilde{\underline{T}}^f = \tilde{\underline{T}}^f_r - c^f_r$, and $\underline{T}^f = \underline{T}^f_r - c^f_r$, $\forall f \in \mathcal{F}, r \in \mathcal{R}_f$, where c^f_r is a constant representing the travel time of flight f to runway r from its origin point in our system. Hence $Z^f = \tilde{\underline{T}}^f_r - \underline{T}^f_r$. In other words, uncertainty in a flight's earliest possible touchdown/takeoff time is determined by the uncertainty in its earliest possible release time into the system.

We assume that the random variables Z^f are identically distributed for each flight type $\boldsymbol{i}$, and further, that they are independent for each flight f. The latter assumption is justifiable since our time horizon is short, so aircraft-specific delays will typically not be transferred from one flight to another during the optimization period. We shall hence denote these random variables by $Z^{\boldsymbol{i}}$, where $\boldsymbol{i} = \boldsymbol{c}_f$. Indeed, the distribution of these random variables can be directly observed by monitoring daily operations at an airport.

We then define the random variables

$$\begin{aligned} Y^{(1)\boldsymbol{i}}_t &\triangleq \text{the number of flights of type } \boldsymbol{i} \text{ that are ``delayed'' at time } t \\ &= \sum_{f \in \mathcal{F}_{\boldsymbol{i}}} \mathbb{I}^{(1)f}_t, \text{ where } \mathbb{I}^{(1)f}_t \triangleq \begin{cases} 1, & \text{if } \underline{T}^f \le t < \tilde{\underline{T}}^f; \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Then

$$
\begin{aligned}
P(\mathbb{I}_t^{(1)f}) &= P(\underline{T}^f \le t < \tilde{\underline{T}}^f) \\
&= \begin{cases} P(Z^{\boldsymbol{i}} > t - \underline{T}^f), & \text{if } t \ge \underline{T}^f; \\ 0, & \text{otherwise,} \end{cases} \\
&= \begin{cases} 1 - \Phi_{Z^{\boldsymbol{i}}}(t - \underline{T}^f), & \text{if } t \ge \underline{T}^f; \\ 0, & \text{otherwise,} \end{cases}
\end{aligned}
$$

where $\Phi_{Z^{\boldsymbol{i}}}$ is the cumulative distribution function of $Z^{\boldsymbol{i}}$, observed in the historic data.

We can define in an analogous fashion

$$Y_t^{(2)\boldsymbol{i}} \triangleq \text{the number of flights of type } \boldsymbol{i} \text{ that are "early" at time } t.$$

Therefore $Y_t^{(1)\boldsymbol{i}}$ is the sum of $|\,\mathcal{F}_{\boldsymbol{i}}|_{\{t \ge \underline{T}^f\}}\,|$ independent bernoulli random variables $\mathbb{I}_t^{(1)f}$ with $p_t^f = 1 - \Phi_{Z^{\boldsymbol{i}}}(t - \underline{T}^f)$, and $Y_t^{(2)\boldsymbol{i}}$ is the sum of $|\,\mathcal{F}_{\boldsymbol{i}}|_{\{t < \underline{T}^f\}}\,|$ independent bernoulli random variables $\mathbb{I}_t^{(2)f}$ with $p_t^f = \Phi_{Z^{\boldsymbol{i}}}(t - \underline{T}^f)$.

Hence we have

$$Y_t^{\boldsymbol{i}} \triangleq Y_t^{(1)\boldsymbol{i}} - Y_t^{(2)\boldsymbol{i}}$$

fewer flights of type $\boldsymbol{i}$ available for release into the system at time t than expected based on flights' nominal release times, where $Y_t^{(1)\boldsymbol{i}}$ and $Y_t^{(2)\boldsymbol{i}}$ each have a so-called *poisson-binomial* distribution.

Computing $N_{\boldsymbol{it}}^{\alpha}$

Now that we have modeled the relevant uncertainty with the above random variables, we relate these to our robust optimization problem RP4-1. In the above presentation of RP4-1 and the ensuing Proposition 4.1, we refrained from defining the right-hand sides $N_{\boldsymbol{it}}^{\alpha}$ of Constraints (4.2b). Indeed, it is natural that these should equal $\max\{0, \Phi_{Y_t^{\boldsymbol{i}}}^{-1}(1-\alpha)\}$, where $\Phi_{Y_t^{\boldsymbol{i}}}^{-1}(1-\alpha)$ is the $100(1-\alpha)^{\text{th}}$ percentile of the distribution of $Y_t^{\boldsymbol{i}}$. In this way, in specifying a parameter α, we are choosing to protect up to the $100(1-\alpha)^{\text{th}}$ percentile of the distribution of delays at time t. In order to compute these values, we need to observe the historic distribution of the random variables $Z^{\boldsymbol{i}}$.

From the data we have available, realizations of these random variables were taken to be as follows:

- for departures, (actual gate out time) – (scheduled gate out time);
- for arrivals, (actual touchdown time) – (scheduled touchdown time).

Here, we used the ASPM data available to us, which gives runway off/on and gate out/in times. The issues of potential concern regarding this approach are threefold: i) we are assuming that a departure's actual gate out time is its actual earliest possible one; ii) we are approximating the delay in the earliest possible time of arrivals at their arrival fix by the delay in their touchdown time; and iii) these delays use scheduled times which were available well in advance of the flight time.

We believe the first two issues above not to be significant sources of error, however the third one could drastically over-estimate the delays in flight release times for which we need to be prepared. This is because in any implementation of our methodology, the scheduled times would be updated immediately before re-solve, taking into account the latest available data. For example, most arrivals would already be en-route and so we would have a much better estimation of their arrival fix time, and so the delay against that estimate would be reduced. As a result, we include a user-specified parameter γ_t^f, which scales the distribution of delays observed in our data. In this way, we have

$$P(\mathbb{I}_t^{(1)f}) = \begin{cases} 1 - \Phi_{Z^{\boldsymbol{i}}}(\frac{t-\underline{T}^f}{\gamma_t^f}), & \text{if } t \geq \underline{T}^f; \\ 0, & \text{otherwise,} \end{cases}$$

and a similar expression for $P(\mathbb{I}_t^{(2)f})$.

The next and final task at hand is that of computing the $100(1-\alpha)^{\text{th}}$ percentile of the distribution of $Y_t^{\boldsymbol{i}}$, which we use to calculate $N_{\boldsymbol{i}t}^{\alpha}$. Indeed, there are efficient algorithms known for computing this – see for example [14]. As the number of bernoulli variables grows large, a more tractable alternative is to use an approximation. Here we can make use of a version of the Central Limit Theorem, by which the sum of independent (and not necessarily identically distributed) random variables converges to a normal random variable, under a certain “Lyapunov” condition.

In our case, it is well-known that for the poisson-binomial distribution of $Y_t^{(1)\boldsymbol{i}}$ and $Y_t^{(2)\boldsymbol{i}}$, the Lyapunov condition is satisfied if

$$\sum_{i=1}^{n} \sigma_i^2 = \sum_{f=1}^{n} p_t^f \left(1 - p_t^f\right) \xrightarrow{n\to\infty} \infty.$$

It then follows that the Lyapunov condition is also satisfied for $Y_t^{\boldsymbol{i}}$ under the same conditions, in which case we have that

$$\frac{Y_t^{\boldsymbol{i}} - \mathbb{E}\left[Y_t^{\boldsymbol{i}}\right]}{\mathrm{sd}\left(Y_t^{\boldsymbol{i}}\right)} = \frac{\sum_{f=1}^{k}\left(\mathbb{I}_t^{(1)f} - p_t^f\right) - \sum_{f=k+1}^{n}\left(\mathbb{I}_t^{(2)f} - p_t^f\right)}{\sqrt{\sum_{f=1}^{n} p_t^f \left(1 - p_t^f\right)}} \xrightarrow{d} \mathcal{N}(0,1),$$

where $k = |\,\mathcal{F}_{\boldsymbol{i}}|_{\{t \geq \underline{T}^f\}}\,|$.

Hence, as the number of flights considered becomes large enough, we can use this normal approximation to compute the values of the right-hand sides of Constraints (4.2b) of RP4-1, $N_{\boldsymbol{it}}^{\alpha}$.

4.2.3 Simulating a Stochastic and Dynamic Environment

Now that we have outlined the robust optimization approach and also how to obtain the necessary uncertainty sets from historic data, we consider how to simulate the implementation of the methodology in a real world environment. Algorithm 4.1 details the approach we take:

Algorithm 4.1. 1. Set $k = 0$ and let $\left[\, t_{(k)},\, t_{(k+1)}\,\right)$ be the current time window, where $t_{(k)}$ is the k^{th} solve time. Values of parameters at iteration k will be indicated with a (k) subscript.

2. **Sample** the realized earliest possible start times $\tilde{\underline{T}}^f$ of flights. Fix and hide these.

3. **While not finished, do**

   (a) **update input data** for optimization problem, including:
   - already fixed flight-runway-time assignments,
   - nominal values $\underline{T}^f_{(k)}$ of unprocessed flights, and
   - robustness parameters $N^{\alpha}_{\boldsymbol{i}t(k)}$.

   (b) **solve** optimization problem with updated data, to determine the ideal processing times of as-yet unprocessed flights.

   (c) **reveal** the realized earliest possible release times which lie in the current time window.

   (d) **process** (i.e. fix the starting time and touchdown/takeoff time, etc, of) certain flights in the current time window by making appropriate assignments of flights to flight type slots, as described below.

   (e) increment k. If termination criteria are met, stop.

Assigning Flights to Flight Type Slots

We now describe how we choose in Step 3(d) of Algorithm 4.1 which flights to process in the interval $\left[t_{(k)}\,,\,t_{(k+1)}\right)$. First we define the set of eligible flights as

$$\mathcal{F}^*_{(k)} = \{f \in \mathcal{F} \,:\, f \text{ is unprocessed and } \tilde{\underline{T}}^f_{(k)} < t_{(k+1)}\}.$$

We shall assign flights to the flight type slots found in the optimization problem RP4-1, given our most recent knowledge of the revealed uncertainty. In order to do this, we shall solve a simple binary integer optimization problem, P4-2$(\boldsymbol{i})_{(k)}$ below, for each flight type $\boldsymbol{i}$:

$$\textbf{P4-2}(\boldsymbol{i})_{(k)} : \max \sum_{f \in \mathcal{F}^*_{\boldsymbol{i}(k)}} \sum_{p \in \mathcal{P}_f} \left[\left(1 - \varepsilon_1 \cdot t_p - \varepsilon_2 \cdot \underline{T}^f_{(k)}\right) y^f_p \right] \tag{4.3a}$$

$$\text{s.t.} \sum_{p \in \mathcal{P}_f} y^f_p \leq 1, \quad \forall f \in \mathcal{F}^*_{\boldsymbol{i}(k)}, \tag{4.3b}$$

$$\sum_{f \in \mathcal{F}^*_{\boldsymbol{i}(k)} : p \in \mathcal{P}_f} y_p^f \leq 1, \quad \forall p \in \cup_{f \in \mathcal{F}^*_{\boldsymbol{i}(k)}} \mathcal{P}_f, \tag{4.3c}$$

$$y_p^f \in \{0,\, 1\}, \quad \forall f \in \mathcal{F}^*_{\boldsymbol{i}(k)},\, p \in \mathcal{P}_f. \tag{4.3d}$$

Here, our decision variables $\boldsymbol{y}$ are defined as follows:

$$y_p^f = \begin{cases} 1, & \text{if flight } f \text{ is assigned to runway slot } p, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\begin{aligned} \mathcal{P}_f &= \text{ the set of runway slots feasible for flight } f, \\ t_p &= \text{ the touchdown/takeoff time of slot } p, \\ \varepsilon_1, \varepsilon_2 &= \text{ some constants such that } 0 < \varepsilon_1, \varepsilon_2 < 1. \end{aligned}$$

Focusing first on the objective function of P4-2$(\boldsymbol{i})_{(k)}$, we can see that if $\varepsilon_1 \cdot t_p + \varepsilon_2 \cdot \underline{T}^f_{(k)}$ is sufficiently small for all slots p, then we maximize the number of slots filled. Also, due to these terms, the model also favors filling earlier slots with flights that have earlier release times, amongst solutions with the same number of slots filled. Constraints (4.3b) state that each flight may be assigned to at most one runway slot. Constraints (4.3c) state that each slot may be filled with at most one flight.

Observe that P4-2$(\boldsymbol{i})_{(k)}$ is a maximum assignment problem and therefore we can solve it as a linear program, and indeed a very small one. As a result, solutions are instantaneous, fitting in with the required time scale of the decision-making.

Finally, we remark that this assignment problem could be modified in numerous ways to achieve a desired objective, for example one might choose to place more emphasis on filling slots with flights which have been waiting longer, in the spirit of fairness. This, however, will not be our focus.

4.2.4 Computational Experience

We now implement the above simulation scheme using historic data at DFW, and analyze the results. In particular, in each simulation we consider a fixed time horizon and a set of flights. The simulations are terminated when the end of the time horizon is reached, at which point each of these flights will either have taken off/landed or not. We repeat the simulations multiple times and record the results.

Figures 4-1 and 4-2 show histograms of the simulated flight delays both for deterministic and robust optimization policies. Here, a flight's runway delay is defined as the difference between its actual (simulated) touchdown/takeoff time and its actual earliest possible touchdown/takeoff time. Note that Figure 4-2 shows "worst case" simulations, where we have sampled from between the 70th and 100th percentiles of the distribution of delays. This might sometimes be applicable when bad weather strikes and many flights are delayed simultaneously.

We can see that the deterministic optimization results in many flights having very low delay. However, the tail of the distribution is large, since many slots were not filled and hence many flights not processed before the end of the time horizon. In the robust optimization, we can see that around the median, the distribution of delays has been slightly right-shifted, with on average slightly higher delay amongst processed flights. However, much more of the flights were processed by the end of the simulation time window, greatly reducing the tail of the distribution. The same phenomenon is seen in Figure 4-2, except it is much more pronounced, as expected.

These observations are reinforced by Tables 4.1 and 4.2, which present the corresponding mean and standard deviations, as well as various percentiles of the distributions of delays. In particular, we can again see that while the median is typically better for deterministic policies, the robust optimization does better near the tails. For example, in Table 4.1, at least 75% of flights have a lower delay under the deterministic optimization policy than under the robust optimization policy, but the most delayed 10% of flights receive significantly less delay under the robust policies.

Another observation we can make from Tables 4.1 and 4.2 is that when the flight

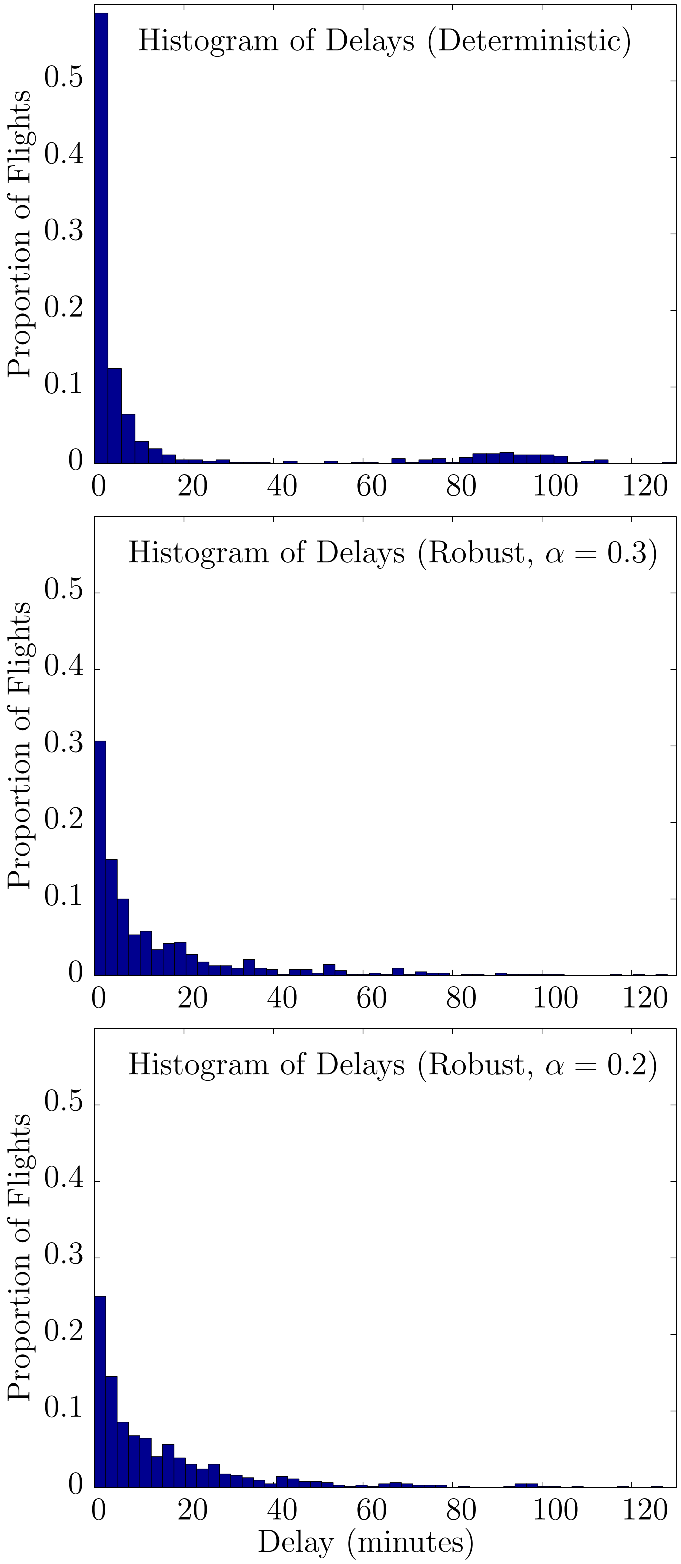

Figure 4-1: Histograms of simulated runway delays at DFW. Here there is no re-solve.

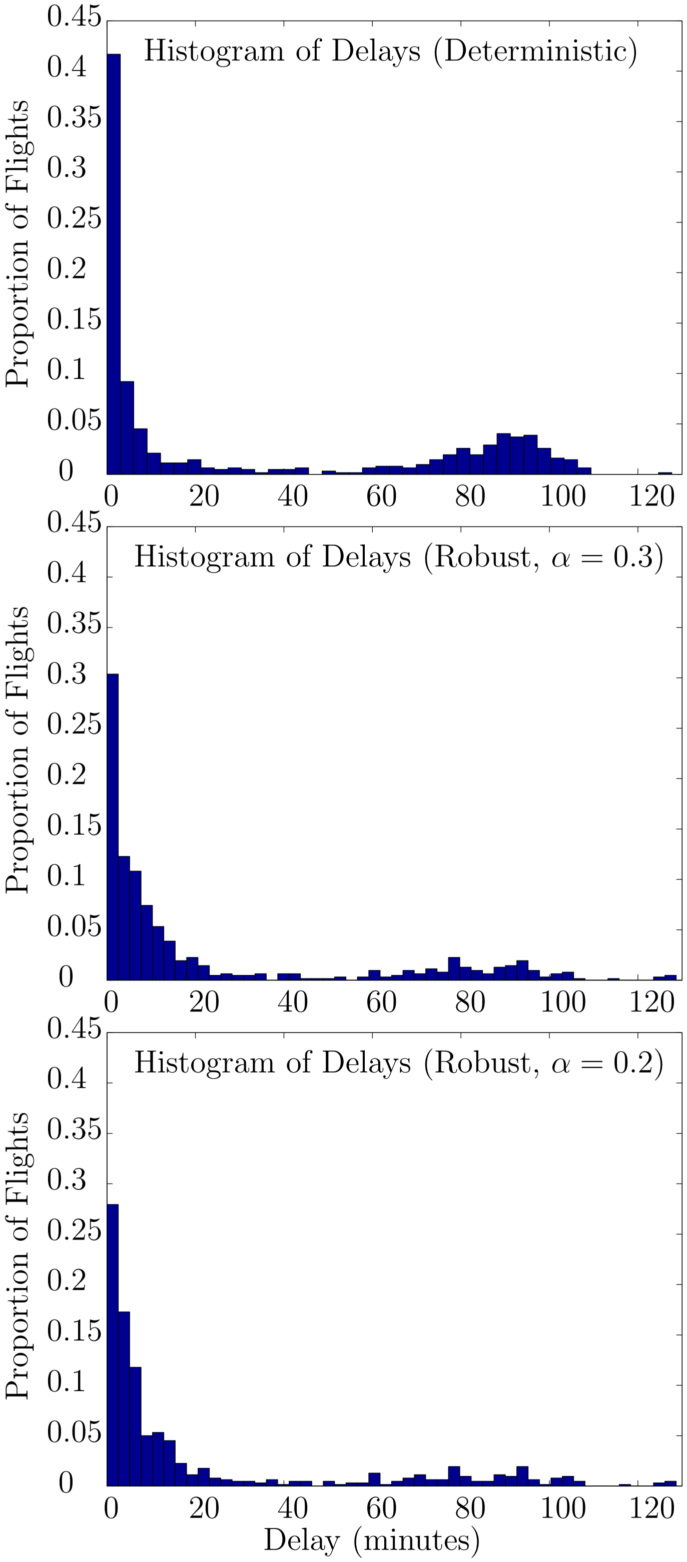

Figure 4-2: Histograms of simulated runway delays at DFW. Here we sample from between the 70th and 100th percentiles of the distribution, and there is no re-solve.

		Default Flight Mix			Even Flight Mix		
		Deterministic	**Robust**		**Deterministic**	**Robust**	
	$\boldsymbol{\alpha}$	—	0.3	0.2	—	0.3	0.2
	Mean	15.8	14.2	15.7	18.9	16.9	17.5
	S.D.	31.7	20.2	20.2	31.9	18.3	15.7
Percentiles	**50th**	2.0	6.0	8.0	4.7	11.0	14.0
	75th	8.0	18.0	20.4	14.4	22.3	25.0
	90th	84.7	38.4	41.0	88.1	39.0	37.0
	95th	96.0	56.0	59.3	101.3	57.4	49.1
	Max.	151.0	128.0	127.0	122.0	103.7	80.3

Table 4.1: Statistics of simulated runway delays for deterministic and robust optimization policies, varying flight mix and robust α value at DFW. Here there is no re-solve.

		Default Flight Mix			Even Flight Mix		
		Deterministic	**Robust**		**Deterministic**	**Robust**	
	$\boldsymbol{\alpha}$	—	0.3	0.2	—	0.3	0.2
	Mean	34.3	22.5	21.3	37.1	24.3	19.2
	S.D.	41.6	32.5	31.5	40.6	30.2	24.6
Percentiles	**50th**	6.0	6.7	6.3	12.3	10.3	10.3
	75th	81.5	22.2	19.7	83.7	25.2	19.7
	90th	95.0	83.0	79.5	97.3	81.1	64.1
	95th	100.7	93.4	93.7	103.5	92.3	82.0
	Max.	152.0	128.7	128.7	122.7	110.7	110.7

Table 4.2: Statistics of simulated runway delays for deterministic and robust optimization policies, varying flight mix and robust α value at DFW. Here we sample from between the 70th and 100th percentiles of the distribution, and there is no re-solve.

		No re-solves			**One re-solve**		
		Deterministic	**Robust**		**Deterministic**	**Robust**	
	$\boldsymbol{\alpha}$	—	0.3	0.2	—	0.3	0.2
	Mean	15.8	14.2	15.7	7.1	11.8	13.3
	S.D.	31.7	20.2	20.2	13.2	14.2	14.9
Percentiles	**50th**	2.0	6.0	8.0	2.0	6.0	7.7
	75th	8.0	18.0	20.4	7.3	17.1	19.1
	90th	84.7	38.4	41.0	18.4	32.0	34.7
	95th	96.0	56.0	59.3	29.4	45.1	44.7
	Max.	151.0	128.0	127.0	101.0	72.0	72.7

Table 4.3: Effect of re-solving on performance. This table shows statistics of simulated runway delays for deterministic and robust optimization policies, and varying robust α value at DFW under the default flight mix.

mix was modified to be more diverse (i.e., equal proportions of Small, Heavy, etc.), delays worsened. This makes sense, in that when the flight mix is more diverse, less flights which have been released early are eligible to fill an empty flight type slot which was intended for a flight which turned out to be released late. This phenomenon was also observed in stochastic runway scheduling approach of [41]. At the same time, the difference in the distribution of delays between the robust and deterministic approaches was also slightly exacerbated, as could be expected.

Now we analyze with Tables 4.3 and 4.4 the effect of introducing a re-solve of the optimization problem mid-horizon. In particular, we can see that with the addition of a re-solve there is a significant reduction in both the mean and variance of delays across all policies. This is primarily due to the largest delays being greatly reduced, as indicated by the percentiles. This reduces the difference between the robust and deterministic approaches. Indeed, in the case where we sample across the entire distribution, the deterministic approach performs better all round (except for in terms of maximum delay). However, under the "worst case" simulation, we still see the same relationship between the delays of the robust and deterministic approaches with one re-solve as with no re-solves, even if the difference between the two is reduced.

Finally, in Tables 4.5 and 4.6, we present a different quantity to previous ta-

		No re-solves			One re-solve		
		Deterministic	**Robust**		**Deterministic**	**Robust**	
	$\boldsymbol{\alpha}$	—	0.3	0.2	—	0.3	0.2
	Mean	34.3	22.5	21.3	11.4	10.1	9.8
	S.D.	41.6	32.5	31.5	15.9	12.9	11.9
Percentiles	**50th**	6.0	6.7	6.3	3.7	5.7	6.0
	75th	81.5	22.2	19.7	16.3	12.7	12.3
	90th	95.0	83.0	79.5	31.3	26.1	24.1
	95th	100.7	93.4	93.7	51.1	39.4	33.7
	Max.	152.0	128.7	128.7	81.0	73.7	67.3

Table 4.4: Effect of re-solving on performance. This table shows statistics of simulated runway delays for deterministic and robust optimization policies, and varying robust α value at DFW under the default flight mix. Here we sample from between the 70th and 100th percentiles of the distribution.

bles: the delays of flights' touchdown/takeoff times in the simulations when measured against the planned touchdown/takeoff times from the relevant IP. The purpose of these tables is to analyze the predictive power of the IP solution of what actually occurs in practice. Indeed, it is clear that the robust optimization results in an IP solution which is much closer to what is achievable in practice than in the deterministic case, as evidenced by the lower values across the board in Tables 4.5 and 4.6.

In summary, we have observed i) that the robust optimization policies can result in significantly lower delays to the most delayed flights, when compared to the deterministic optimization policies (particularly when the realizations of randomness are unfavorable); ii) however, this benefit comes at the cost of a slight increase to the delays of most flights (i.e., of the flights which do not experience high delay); iii) that re-solving the IP mid-horizon reduces the difference between the robust and deterministic approaches, with the robust approach performing better only when the realizations of randomness are worse; and iv) that the robust optimization approach is more predictable, in the sense that, in the simulation environment, it performs closer to the policy indicated by the IP solution than the deterministic approach, even if the robust policies are more conservative ones.

		No re-solves			**One re-solve**		
		Deterministic	**Robust**		**Deterministic**	**Robust**	
	$\boldsymbol{\alpha}$	—	0.3	0.2	—	0.3	0.2
	Mean	15.6	7.9	8.6	6.8	5.5	6.2
	S.D.	31.7	18.1	18.7	13.2	11.2	11.3
Percentiles	**50th**	2.0	0.0	0.3	2.0	0.0	0.0
	75th	7.7	5.4	6.7	6.8	5.0	7.3
	90th	84.1	26.9	25.4	17.7	17.7	21.7
	95th	95.7	48.4	50.0	29.0	34.4	30.7
	Max.	151.0	126.3	122.0	100.3	66.0	68.3

Table 4.5: Predictability of practical performance from IP solution. This table shows statistics of simulated runway delays, *measured against the IP solution*, for deterministic and robust optimization policies, and varying robust α value at DFW under the default flight mix.

		No re-solves			**One re-solve**		
		Deterministic	**Robust**		**Deterministic**	**Robust**	
	$\boldsymbol{\alpha}$	—	0.3	0.2	—	0.3	0.2
	Mean	34.0	17.3	15.9	11.1	5.3	4.8
	S.D.	41.6	31.9	30.8	15.9	11.8	10.4
Percentiles	**50th**	5.3	0.0	0.0	3.7	0.0	0.0
	75th	81.3	11.0	9.0	15.7	5.0	4.2
	90th	94.7	77.1	75.0	30.7	18.1	16.1
	95th	100.1	91.4	90.7	50.7	30.0	28.0
	Max.	152.0	127.0	128.0	81.0	68.3	67.3

Table 4.6: Predictability of practical performance from IP solution. This table shows statistics of simulated runway delays, *measured against the IP solution*, for deterministic and robust optimization policies, and varying robust α value at DFW under the default flight mix. Here we sample from between the 70th and 100th percentiles of the distribution.

4.3 Incorporating Uncertainty in Runway Availability

According to FAA document 8400-9 [18], the factors which influence runway usability are wind-shear or thunderstorms, visibility, runway braking effectiveness (as influenced by slippery conditions), winds and other safety factors. Under clear and dry conditions, for a runway to be usable, the cross-wind component of wind may not exceed 20 knots (including gusts), and the tailwind component may not exceed 7 knots. We note that all of these factors would generally shut down either all or none of the runways at an airport, except the wind conditions and visibility (some runways are equipped for very low visibility, while others are not). In the former case, we can do nothing to get around the reduced (zero) capacity, hence we focus here on the influence of wind and visibility on runway availability.

Notice that since there are a small number of runways of substantially different orientation, small perturbations in the wind will typically not change the availability of runways. As a result, there will typically be only a few possible runway-availability combinations corresponding to any wind and visibility forecast. We shall take advantage of this fact to produce a stochastic optimization problem with similar computational tractability to that of RP4-1.

In particular, we shall fix in our model some time $\check{T} \approx T/2$ such that during time periods $1, \ldots, \check{T}$, runway availability stays constant with high probability (given our short time horizon). Following on from our above discussion, it is a fair assumption that in practice only a few scenarios would be given serious consideration, unless the wind direction were truly erratic: i) no change in runway availability occurs, ii) at some time $t^* > \check{T}$, one of a few possible runway availability scenarios eventuates.

As mentioned above, we shall propose a two-stage stochastic optimization approach. Given the small number of possible scenarios, computational tractability should not be affected too adversely. These models involve two stages of decision-making, which we shall denote by decision variables $\boldsymbol{x}$ and $\boldsymbol{y}$. In the first stage, we make a set of decisions $\boldsymbol{x}$ before the realization of some uncertainty $\boldsymbol{\xi} \in \Xi$, after which we make a second set of decisions $\boldsymbol{y}$. Our two stages are then:

- Stage I: time periods $\{1, \ldots, \check{T}\}$, and
- Stage II: time periods $\{\check{T}+1, \ldots, T\}$.

We remind the reader using notation from [39] that a two-stage stochastic IP takes the form:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & \boldsymbol{c}'\boldsymbol{x} + \mathbb{E}_{\boldsymbol{\xi}\in\Xi}[Q(\boldsymbol{x},\boldsymbol{\xi})], \text{ where} \\ Q(\boldsymbol{x},\boldsymbol{\xi}) \quad & = \min\nolimits_{\boldsymbol{y}(\boldsymbol{\xi})} \quad \boldsymbol{d}'(\boldsymbol{\xi})\boldsymbol{y}(\boldsymbol{\xi}) \\ & \qquad \text{s.t.} \qquad E(\boldsymbol{\xi})\boldsymbol{x} + F(\boldsymbol{\xi})\boldsymbol{y}(\boldsymbol{\xi}) \le \boldsymbol{h}(\boldsymbol{\xi}). \end{aligned}$$

In our case, where Ξ is a discrete set of scenarios of low cardinality, this becomes

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{|\Xi|}} \quad & \boldsymbol{c}'\boldsymbol{x} + \sum_{k=1}^{|\Xi|} \boldsymbol{d}_k'\boldsymbol{y}_k \\ \text{s.t.} \quad & \boldsymbol{x} \in \mathcal{X}, \\ & E_k\boldsymbol{x} + F_k\boldsymbol{y}_k \le \boldsymbol{h}_k, \quad \forall k \in \{1, \ldots, |\Xi|\}. \end{aligned}$$

We can see that this adds approximately $\frac{T-\check{T}}{T}(|\Xi|-1)N$ variables to our formulation P3-1, which is equal to $N/2$ for $\check{T} = T/2$ and $|\Xi| = 2$, where N is the original number of variables.

4.3.1 A Two-Stage Stochastic Robust Optimization Problem

We now specify the full details of a two-stage stochastic robust optimization approach, which builds on RP4-1 to incorporate uncertainty in runway availability.

Data

The following is a list of data required in addition to that of RP4-1:

$\check{T}(\in \mathcal{T})$ = the final time period in the first stage of optimization, after which our different scenarios can occur;

$\mathcal{S}$ = the set of possible scenarios to take place from time $\check{T}+1$ to time T;

p_s = the probability that scenario s will occur;

$\mathcal{U}_t^s$ = those runways which cannot be used at time t in scenario s;

α = a user-chosen parameter indicating the degree to which we wish to protect against delays to flight release times;

$\Phi_{Z^{\boldsymbol{i}}}$ = the historically observed cumulative distribution function of the random variable $Z^{\boldsymbol{i}}$ defined above, for which we do not need an analytic expression;

$N_{\boldsymbol{i}t}^{\alpha}$ = $\max\{0, \Phi^{-1}_{Y_t^{\boldsymbol{i}}}(1-\alpha)\}$, where $\Phi^{-1}_{Y_t^{\boldsymbol{i}}}(1-\alpha)$ is the $100(1-\alpha)^{\text{th}}$ percentile of the distribution of $Y_t^{\boldsymbol{i}}$, which is related to $Z^{\boldsymbol{i}}$ as mentioned above. The details of how to obtain these values was outlined above in Section 4.2.2.

Decision Variables

Our decision variables are as follows:

$$\omega^s_{\boldsymbol{k}t} = \begin{cases} 1, & \text{if configuration } \boldsymbol{k} \text{ is active at time } t \text{ in scenario } s, \\ 0, & \text{otherwise;} \end{cases}$$

$$\varphi^{fs}_{rt} = \begin{cases} 1, & \text{if } f \text{ assigned to runway } r \text{ \& could normally arrive by } t \text{ in scenario } s, \\ 0, & \text{otherwise;} \end{cases}$$

$$\psi^{\boldsymbol{i}s}_{rt} = \begin{cases} 1, & \text{if a flight of type } \boldsymbol{i} \text{ arrives at runway } r \text{ at time } t \text{ in scenario } s, \\ 0, & \text{otherwise;} \end{cases}$$

$$\chi^s_t = \begin{cases} 1, & \text{if a change of configuration occurs at time } t \text{ in scenario } s, \\ 0, & \text{otherwise;} \end{cases}$$

$\Gamma^{\boldsymbol{i}s}_{rt} =$ the protection level against delays to flights of type $\boldsymbol{i}$ assigned to runway r by time t (whose physical meaning is as in RP4-1) in scenario s.

Note that we have added another index to all variables corresponding to the scenario, and we have also added a time index to the $\boldsymbol{\varphi}$ variables and defined them to be "by" variables, in the spirit of [9]. This change will have very little effect on the model tractability, owing to their monotonic nature, and could in fact be used in P3-1 if desired. It is important to realize that the $\boldsymbol{\varphi}$ variables do not here give us a hold on the precise touchdown/takeoff time of each flight when $\boldsymbol{\Gamma} \neq \mathbf{0}$ (as is typically the case here), although they would if used in P3-1. Instead, what they represent here is the assigned runway and touchdown/takeoff time for each flight if all flights' release times took their nominal values.

Objective Function

Consider the function (4.4) below, which represents a weighted summation of flight delay costs:

$$
\begin{aligned}
\Psi_s \triangleq \sum_{\boldsymbol{i}\in\mathcal{C}} \left(\beta_D^G . \mathbb{I}_{\{\boldsymbol{i}\in\mathcal{C}_D\}} + \beta_A^A . \mathbb{I}_{\{\boldsymbol{i}\in\mathcal{C}_A\}}\right) \sum_{r\in\mathcal{R}_{\boldsymbol{i}}} \sum_{t\in\mathcal{T}} t\psi_{rt}^{\boldsymbol{i}s} - \sum_{f\in\mathcal{F}} \left(\beta_D^G . \mathbb{I}_{\{f\in\mathcal{F}_D\}} + \beta_A^A . \mathbb{I}_{\{f\in\mathcal{F}_A\}}\right) \underline{T}_{o_f}^f \\
+ \sum_{f\in\mathcal{F}} \sum_{r\in\mathcal{R}_f} \left(\beta_D^A . \mathbb{I}_{\{f\in\mathcal{F}_D\}} + \beta_A^G . \mathbb{I}_{\{f\in\mathcal{F}_A\}}\right) \cdot d_r^f \varphi_{r\overline{T}_r^f}^{fs} \\
- \left(\beta_D^G - \beta_G\right) \cdot \left(\sum_{\boldsymbol{i}\in\mathcal{C}_D} \sum_{r\in\mathcal{R}_{\boldsymbol{i}}} \sum_{t\in\mathcal{T}} t\psi_{rt}^{\boldsymbol{i}s} - \sum_{f\in\mathcal{F}_D} \sum_{r\in\mathcal{R}_f} \underline{T}_r^f \varphi_{r\overline{T}_r^f}^{fs} \right) \\
+ K \cdot \sum_{t\in\mathcal{T}} \chi_t^s - C \cdot \sum_{\boldsymbol{i}\in\mathcal{C}} \sum_{r\in\mathcal{R}_{\boldsymbol{i}}} \sum_{t\in\mathcal{T}} \Gamma_{rt}^{\boldsymbol{i}s}. \qquad (4.4)
\end{aligned}
$$

This expression is again a very simple extension to the deterministic case of P3-1, with the notable addition which penalizes the $\boldsymbol{\Gamma}$ parameters as in the objective of RP4-1.

There are then in fact two related objective functions we can choose to minimize, each of which uses the above quantities Ψ_s. The first objective, which we shall use in our experiments, is to minimize the expectation over the scenarios:

$$
\min \Psi, \text{ where } \Psi = \sum_{s\in\mathcal{S}} p_s \Psi_s,
$$

while the second possible objective function is to minimize the worst case value:

$$\min \Psi, \text{ where } \Psi = \max_{s \in \mathcal{S}} \Psi_s,$$

in which case we add the to the model the constraints $\Psi \geq \Psi_s,\ \forall s \in \mathcal{S}$.

The Stochastic Robust Optimization Model

Finally, we have the following stochastic robust integer optimization problem, which protects against uncertainty in both flights' earliest possible touchdown/takeoff times, and the availability of runways for use:

$$\begin{aligned}
\textbf{SRP4-3:}\ \min \quad & \Psi \\
\text{s.t.} \quad & (\boldsymbol{\psi}_t^{s_1\prime}, \boldsymbol{\omega}_t^{s_1\prime}, \chi_t^{s_1}) - (\boldsymbol{\psi}_t^{s_2\prime}, \boldsymbol{\omega}_t^{s_2\prime}, \chi_t^{s_2}) = \mathbf{0}', \\
& \qquad \forall s_1 \neq s_2 \in \mathcal{S},\ t \in \{1, \ldots, \check{T}\}, && \text{(4.5a)} \\
& \varphi_{rt}^{f s_1} - \varphi_{rt}^{f s_2} = 0, \\
& \qquad \forall s_1 \neq s_2 \in \mathcal{S},\ f \in \mathcal{F},\ r \in \mathcal{R}_f,\ t \in \mathcal{T}_r^f \cap \{1, \ldots \check{T}\}, && \text{(4.5b)} \\
& \sum_{\boldsymbol{k} \in \mathcal{K}} \omega_{\boldsymbol{k}t}^s = 1, \quad \forall s \in \mathcal{S},\ t \in \mathcal{T}, && \text{(4.5c)} \\
& \psi_{rt}^{\boldsymbol{i}s} = 0, \quad \forall s \in \mathcal{S},\ \boldsymbol{i} \in \mathcal{C},\ r \in \mathcal{U}_t^s,\ t \in \mathcal{T}, && \text{(4.5d)} \\
& \psi_{r,t-h}^{\boldsymbol{i}s} + \psi_{rt}^{\boldsymbol{j}s} \leq 1, \quad \forall s \in \mathcal{S},\ \boldsymbol{i}, \boldsymbol{j} \in \mathcal{C},\ r \in \mathcal{R}_{\boldsymbol{i}} \cap \mathcal{R}_{\boldsymbol{j}}, \\
& \qquad h \in \{1, \ldots, \min\{s_{\boldsymbol{ij}}^r - 1, t - 1\}\},\ t \in \mathcal{T}, && \text{(4.5e)} \\
& \psi_{r,t-h}^{\boldsymbol{i}s} + \psi_{r',t}^{\boldsymbol{j}s} \leq 1, \quad \forall s \in \mathcal{S},\ \boldsymbol{i}, \boldsymbol{j} \in \mathcal{C},\ (r, r') \in (\mathcal{R}_{\boldsymbol{i}} \times \mathcal{R}_{\boldsymbol{j}}) \cap \mathcal{V}, \\
& \qquad h \in \{0, \ldots, \min\{s_{\boldsymbol{ij}}^{(r,r')} - 1, t - 1\}\},\ t \in \mathcal{T}, && \text{(4.5f)} \\
& \sum_{\boldsymbol{i} \in \mathcal{C} \,:\, r \in \mathcal{R}_{\boldsymbol{i}}} \psi_{rt}^{\boldsymbol{i}s} \leq 1, \quad \forall s \in \mathcal{S},\ r \in \mathcal{R},\ t \in \mathcal{T}, && \text{(4.5g)} \\
& \psi_{rt}^{\boldsymbol{i}s} + \omega_{\boldsymbol{k}t}^s \leq 1, \quad \forall s \in \mathcal{S},\ t \in \mathcal{T},\ \boldsymbol{k} \in \mathcal{K},\ r \in \mathcal{R}_{\boldsymbol{k}}, \boldsymbol{i} \in \bar{\mathcal{I}}_{r\boldsymbol{k}} \,:\, r \in \mathcal{R}_{\boldsymbol{i}}, \\
& && \text{(4.5h)}
\end{aligned}$$

$$\psi^{\boldsymbol{i}s}_{rt} - \sum_{\substack{\boldsymbol{k}\in\mathcal{K}\,:\,r\in\mathcal{R}_{\boldsymbol{k}},\\ \boldsymbol{i}\in\mathcal{I}_{r\boldsymbol{k}}}} \omega^{s}_{\boldsymbol{k},t+h} \leq 0, \quad \forall s \in \mathcal{S},\, \boldsymbol{i} \in \mathcal{C},\, r \in \mathcal{R}_{\boldsymbol{i}},$$

$$h \in \{0, \ldots, \min\{l^r_{\boldsymbol{i}} - 1, T - t\}\},\, t \in \mathcal{T}, \tag{4.5i}$$

$$\varphi^{fs}_{r,t-1} - \varphi^{fs}_{rt} \leq 0, \quad \forall s \in \mathcal{S},\, f \in \mathcal{F},\, r \in \mathcal{R}_f,\, t \in \mathcal{T}^f_r \setminus \{\underline{T}^f_r\}, \tag{4.5j}$$

$$\sum_{r\in\mathcal{R}_f} \varphi^{fs}_{r\overline{T}^f_r} = 1, \quad \forall s \in \mathcal{S},\, f \in \mathcal{F}, \tag{4.5k}$$

$$\sum_{\substack{f\in\mathcal{F}_{\boldsymbol{i}}\,:\,r\in\mathcal{R}_f,\\ t\geq\overline{T}^f_r - l^r_{\boldsymbol{i}}+1}} \varphi^{fs}_{rt} \leq \sum_{\tau=1}^{t} \psi^{\boldsymbol{i}s}_{r\tau} \leq \sum_{\substack{f\in\mathcal{F}_{\boldsymbol{i}}\,:\,r\in\mathcal{R}_f,\\ t\geq\underline{T}^f_r}} \varphi^{fs}_{rt} - \Gamma^{\boldsymbol{i}s}_{rt},$$

$$\forall s \in \mathcal{S},\, \boldsymbol{i} \in \mathcal{C},\, r \in \mathcal{R}_{\boldsymbol{i}},\, t \in \mathcal{T}, \tag{4.5l}$$

$$\chi^s_t - \omega^s_{\boldsymbol{k}t} + \omega^s_{\boldsymbol{k},t-1} \geq 0, \quad \forall s \in \mathcal{S},\, \boldsymbol{k} \in \mathcal{K},\, t \in \mathcal{T} \setminus \{1\}, \tag{4.5m}$$

$$\Gamma^{\boldsymbol{i}s}_{rt} \leq N^{\alpha s}_{\boldsymbol{i}t}, \quad \forall s \in \mathcal{S},\, \boldsymbol{i} \in \mathcal{C},\, r \in \mathcal{R}_{\boldsymbol{i}},\, t \in \mathcal{T}, \tag{4.5n}$$

$$\omega^s_{\boldsymbol{k}t} \in \{0,1\}, \quad \forall s \in \mathcal{S},\, \boldsymbol{k} \in \mathcal{K},\, t \in \mathcal{T}, \tag{4.5o}$$

$$\varphi^{fs}_{rt} \in \{0,1\}, \quad \forall s \in \mathcal{S},\, f \in \mathcal{F},\, r \in \mathcal{R}_f,\, t \in \mathcal{T}^f_r, \tag{4.5p}$$

$$\psi^{\boldsymbol{i}s}_{rt} \in \{0,1\}, \quad \forall s \in \mathcal{S},\, \boldsymbol{i} \in \mathcal{C},\, r \in \mathcal{R}_{\boldsymbol{i}},\, t \in \mathcal{T}. \tag{4.5q}$$

We first mention that for ease of exposition, we have created $|\mathcal{S}|$ sets of variables which span the entire time horizon. In reality, $|\mathcal{S}| - 1$ of these are defined for times $\check{T}+1, \ldots, T$, and one of them for times $1, \ldots, \check{T}$. As a result of this notation decision, there are many more variables listed above than are actually used in practice, as well as many more non-anticipativity constraints. These are Constraints (4.5a) and (4.5b), and ensure that the first-stage decisions are made without knowledge of the scenario that will occur in the second stage. Note that all of our data for time periods $1, \ldots, \check{T}$ is identical across all scenarios under this notation scheme.

Constraints (4.5c) to (4.5m) are essentially the same as in P3-1, but defined for every scenario $s \in \mathcal{S}$, with a few exceptions: Constraints (4.5j) enforce the monotonicity in time of the $\boldsymbol{\varphi}$ variables corresponding to their definition. Constraints (4.5l) have the robustness parameters $\boldsymbol{\Gamma}$ added, and Constraints (4.5n) constrain these $\boldsymbol{\Gamma}$, both as in RP4-1.

Probability runways 17L, 17C, 17R, 18L, 18R not available in second stage	0	0.1	0.2	0.5
Configuration used in first stage	South flow	South flow	North flow	North flow

Table 4.7: Effect of considering stochasticity in runway availability on the optimal configuration selection. Runways 17L, 17C, 17R, 18L, 18R form part of the "South flow" configuration at DFW, as show in Appendix B.

4.3.2 Computational Experience

We now present computational results highlighting the effect of the stochastic optimization problem SRP4-3. In our computations here, we shall not include the robust elements of SRP4-3, which were studied above in Section 4.2, focusing instead on uncertainty in runway availability. Firstly, we demonstrate how considering uncertainty in runway availability can affect the configuration in which we operate: Table 4.7 shows the results of the configuration optimization as we increase the probability of second stage infeasibility of runways 17L, 17C, 17R, 18L and 18R at DFW from 0 to 0.1, 0.2, and finally 0.5. We observe that in the deterministic optimization, the optimal configuration to be used throughout the time horizon is the "South flow" one. However, as the probability of these runways, which belong to the South flow configuration, becoming unavailable in the second stage increases, the optimal solution is changed to use the "North flow" configuration in the first stage instead.

Another benefit of this stochastic optimization is that we have a solution vector for each scenario – if any of these configuration-availability scenarios eventuates, we have a feasible solution to implement, without having to re-solve the optimization. In contrast, when we use a deterministic optimization, if a configuration we are using becomes unavailable at any moment, it is unclear how we would come up with a back-up policy until a re-solve has been implemented.

Next, we assess the tractability of the stochastic model. Tables 4.8 and 4.9 show computation times for two scenarios and three scenarios, respectively. We can observe

Probability runways 17L, 17C, 17R, 18L, 18R not available in second stage	0	0.1	0.2	0.5
Optimality Gap (%)	0	2.4	1.2	0
Comp. Time	1646	1618	1234	1043

Table 4.8: Computational tractability of two-stage stochastic optimization, with 2 scenarios. Solver termination criteria: if within 5% of optimality after 1200s, stop; else stop after 2400s.

Prob. of Scenarios 1 & 2, Respectively	1/0	0.33/0.33	0.5/0.4	0.5/0.1
Optimality Gap (%)	0	5.4	5.0	5.9
Comp. Time	1646	2413	1267	2418

Table 4.9: Computational tractability of two-stage stochastic optimization, with 3 scenarios. Note scenario 1 has all runways available, scenario 2 has runways 17L, 17C, 17R, 18L and 18R unavailable, and scenario 3 has runways 35L, 35C, 35R, 36L and 36R unavailable. Solver termination criteria: if within 5% of optimality after 1200s, stop; else stop after 2400s.

that, surprisingly, there is no significant degradation in running times for two scenarios. On the other hand, there is a slight degradation in the case of three scenarios, typically obtaining solutions between 5–10% of proven optimality after 20 minutes, compared to optimality for the corresponding deterministic optimization.

4.4 Summary

In this chapter, we have considered the effect of uncertainty on a real-world implementation of the methodology of Chapter 3. In particular, we have extended the approach using techniques from robust optimization in order to protect against uncertainty in flights' earliest possible touchdown/takeoff times, and simulated the performance of both the original deterministic approach and the robust approach in a real-world environment. We have observed that re-solving the IP at short intervals can substantially reduce the negative impact of this uncertainty on our optimal policies. Furthermore, the use of robust optimization can provide protection against extreme scenarios, resulting in policies which lead to more predictable results in practice, even though on average their performance may often be worse than the deterministic approach.

Finally, we have also extended the robust optimization approach to form a two-stage stochastic optimization approach which considers the possibility of runways not being available. It was observed that this can potentially result in a different runway configuration selection than under a deterministic optimization. Furthermore, we are furnished with a policy for each of the configuration availability eventualities considered, meaning that we do not have to perform a re-solve or use some other fallback algorithm in the case of catastrophic events. While a negative effect on computation times was seen to exist, this degradation was small enough to conclude that this may be a valid approach in future.

Chapter 5

Concluding Remarks

In this thesis, we have outlined both strategic and tactical optimization models for air traffic flow management at airports. Each of these approaches incorporated all key subproblems in a unified manner, rather than solving them in isolation as is common in the literature. The result is an increased potential for system-optimal solutions. Furthermore, the models have been shown to be computationally tractable in a practical sense, opening the possibility for eventual implementation. Indeed, our approach to solve the AOOP was extended to take into account real world uncertainty, leading to increased reliability in practice.

We advocate an airport-centric approach to optimizing national air traffic, which is a natural one, especially in the United States since often the most critically constrained elements of the air-traffic system are the airports. Moreover, given the efforts of the FAA to transfer airborne delays to ground delays through the use of ground delay programs (GDPs), the importance of optimization at airports further increases. GDPs come into effect when there is inclement weather either en-route or at a flight's destination airport, in which case the FAA reduces that airport's acceptance rate (AAR), and as a result certain arrivals are forced to be held at their origin airport. In this sense, besides having implications at the airport being optimized itself, the work of this thesis can be used to determine AARs and thus affect air traffic on the network level through the use of GDPs.

We note one final important point related to any potential implementation of the methodologies outlined in this thesis – that we have not considered the aspect of fairness between the different agents involved – it is a possibility, for example, that different airlines will be treated differently by a "system-optimal" solution. This represents a key area for future research. Nevertheless, the methodologies we have developed represent a significant step towards improving ATFM at airports.

Appendix A

Separation Calculations

A.1 Separation Rules

The separation rules presented herein were used in this thesis, and were obtained from [16]. First, we denote the velocity of aircraft of type $\boldsymbol{i}$ by $v_{\boldsymbol{i}}$, and the minimum runway occupancy time of flights of type $\boldsymbol{i}$ at runway r by $l^r_{\boldsymbol{i}}$. The data for these parameters is presented in Table A.1.

We then have the following rules for the four cases of separation:

The Arrival-Arrival case.

$$s_{\boldsymbol{ij}} = \begin{cases} \max\left\{l^r_{\boldsymbol{i}}, \frac{r+p_{\boldsymbol{ij}}}{v_{\boldsymbol{j}}} - \frac{r}{v_{\boldsymbol{i}}}\right\}, & v_{\boldsymbol{i}} > v_{\boldsymbol{j}}, \\ \max\left\{l^r_{\boldsymbol{i}}, \frac{p_{\boldsymbol{ij}}}{v_{\boldsymbol{j}}}\right\}, & v_{\boldsymbol{i}} \le v_{\boldsymbol{j}}, \end{cases}$$

Flight Type	Velocity (nmi/hr)	Runway Occupancy Time (s)
Heavy	150	60
Boeing-757	140	60
Large	130	60
Small	110	60

Table A.1: Table of velocities $v_{\boldsymbol{j}}$ and minimum runway occupancy times $l^r_{\boldsymbol{i}}$ by weight class category. [16]

where p_{ij} (shown in Table A.2) is the minimum distance of separation required at all times along the common final approach path of length r.

The Arrival-Departure case.

$$s_{ij} = l_i^r.$$

The Departure-Arrival case.

$$s_{ij} = \max\{l_i^r, \frac{2}{v_j}\}.$$

The Departure-Departure case.

$$s_{ij} = \begin{cases} \min\left\{120, \frac{d_{ij}}{v_i}\right\}, & i = \text{ Heavy/Boeing-757,} \\ \max\left\{l_i^r, \frac{1}{v_i}\right\}, & i = \text{ Large,} \\ \max\left\{l_i^r, \frac{0.75}{v_i}\right\}, & i = \text{ Small,} \end{cases}$$

where d_{ij} is taken from Table A.3.

A.2 Data Used

- r = length of common approach = 5 nautical miles (nmi).
- See Tables A.2 and A.3 for the separation distance requirements.

A.3 Separation Times Used

The above methodology and data gives the separations requirements indicated in Table A.4.

		Trailing Aircraft			
		A-H	**A-757**	**A-L**	**A-S**
Leading Aircraft	**A-H**	4	5	5	5/6*
	A-757	4	4	4	5
	A-L	2.5	2.5	2.5	3/4*
	A-S	2.5	2.5	2.5	2.5

Table A.2: Single runway Arrival-Arrival separation requirements p_{ij}, in nmi. [16] Note * indicates required separation at the runway threshold, while all other separations must apply along the entire common approach path.

		Trailing Aircraft			
		D-H	**D-757**	**D-L**	**D-S**
Leading Aircraft	**D-H**	4	5	5	5
	D-757	4	4	4	5

Table A.3: Single runway Departure-Departure separation requirements d_{ij}, in nmi. [16]

		Trailing Aircraft							
		A-H	**A-757**	**A-L**	**A-S**	**D-H**	**D-757**	**D-L**	**D-S**
Leading Aircraft	**A-H**	96	137	157	207	60	60	60	60
	A-757	96	103	121	199	60	60	60	60
	A-L	60	64	69	123	60	60	60	60
	A-S	60	64	69	82	60	60	60	60
	D-H	60	60	60	60	96	120	120	120
	D-757	60	60	60	60	96	96	111	120
	D-L	60	60	60	60	60	60	60	60
	D-S	60	60	60	60	60	60	60	60

Table A.4: Single-runway separation requirements, in seconds.

Appendix B

Configurations at BOS and DFW

Here we present the feasible configurations we have used at BOS and DFW. Note that there are many subconfigurations of these which are also made possible in our models.

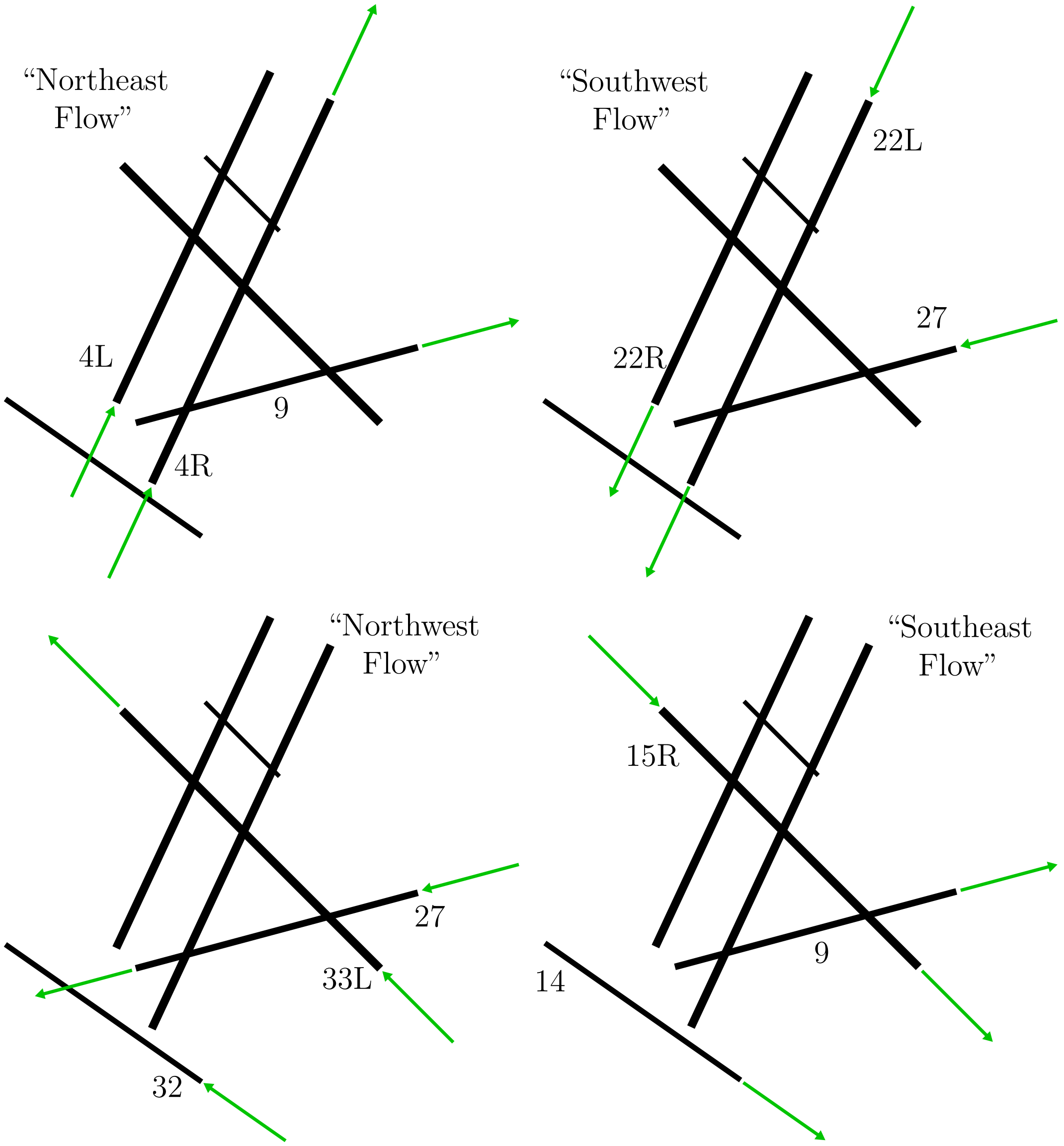

Figure B-1: BOS configurations.

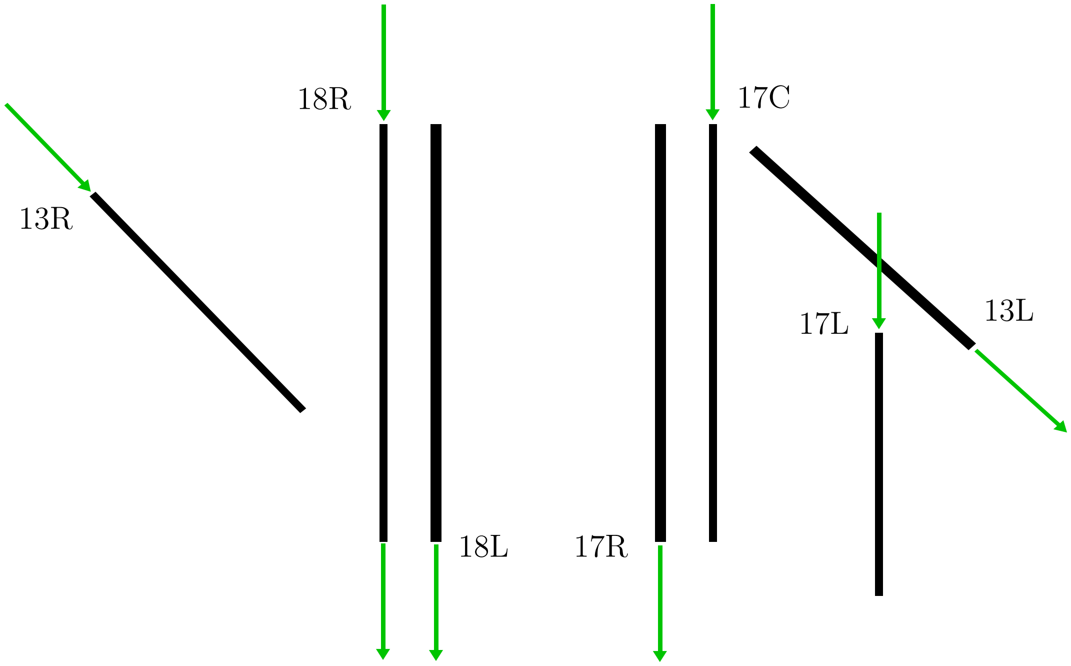

Figure B-2: DFW “South Flow” configuration.

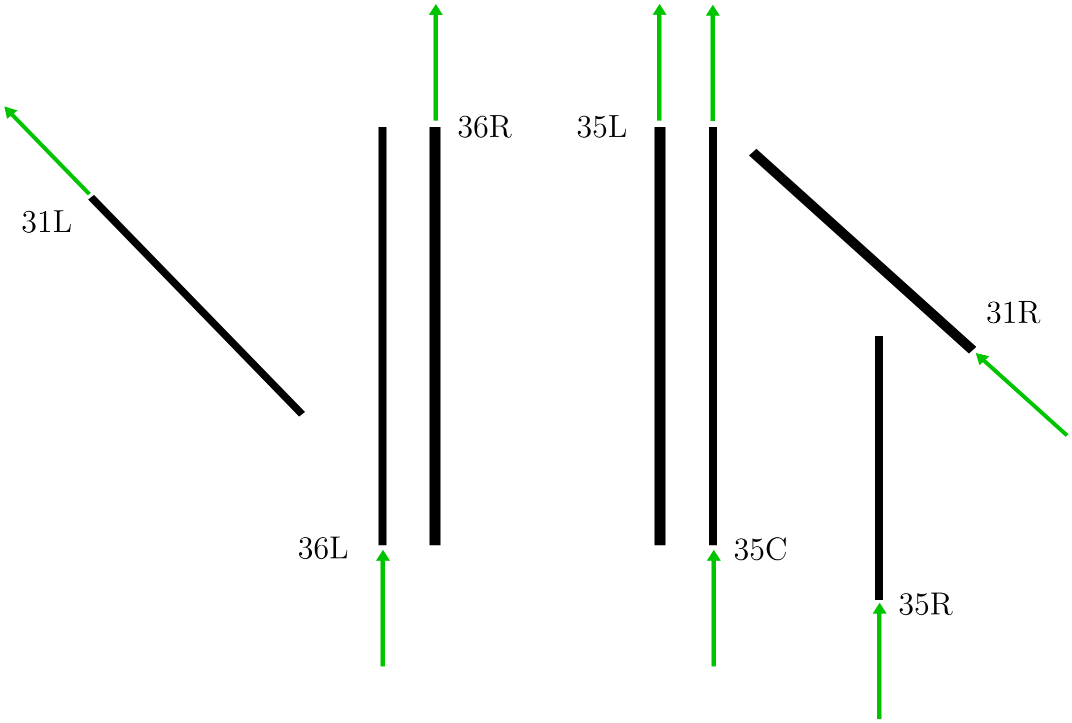

Figure B-3: DFW “North Flow” configuration.

Appendix C

Example Simplified Network for DFW

In order to test our methodology on historical datasets at DFW and BOS, the airport networks were approximated to fit our model. Here, the terminal area and near-terminal airspace were broken down into segments, each forming a node of the simplified network.

Figure C-1 shows the airport diagram of DFW [20], while Figure C-2 shows an example simplified taxiway network (note that the exact one used had more detail, in particular at runway/taxiway crossing points). Figures C-2 – C-5 are presented with permission from Tom Reynolds of MIT Lincoln Laboratory.

Figure C-3 shows a diagram of the DFW arrival and departure fixes, and Figures C-4 and C-5 then show example simplified airspace flight paths along with their lengths, each path being represented by a node of the network.

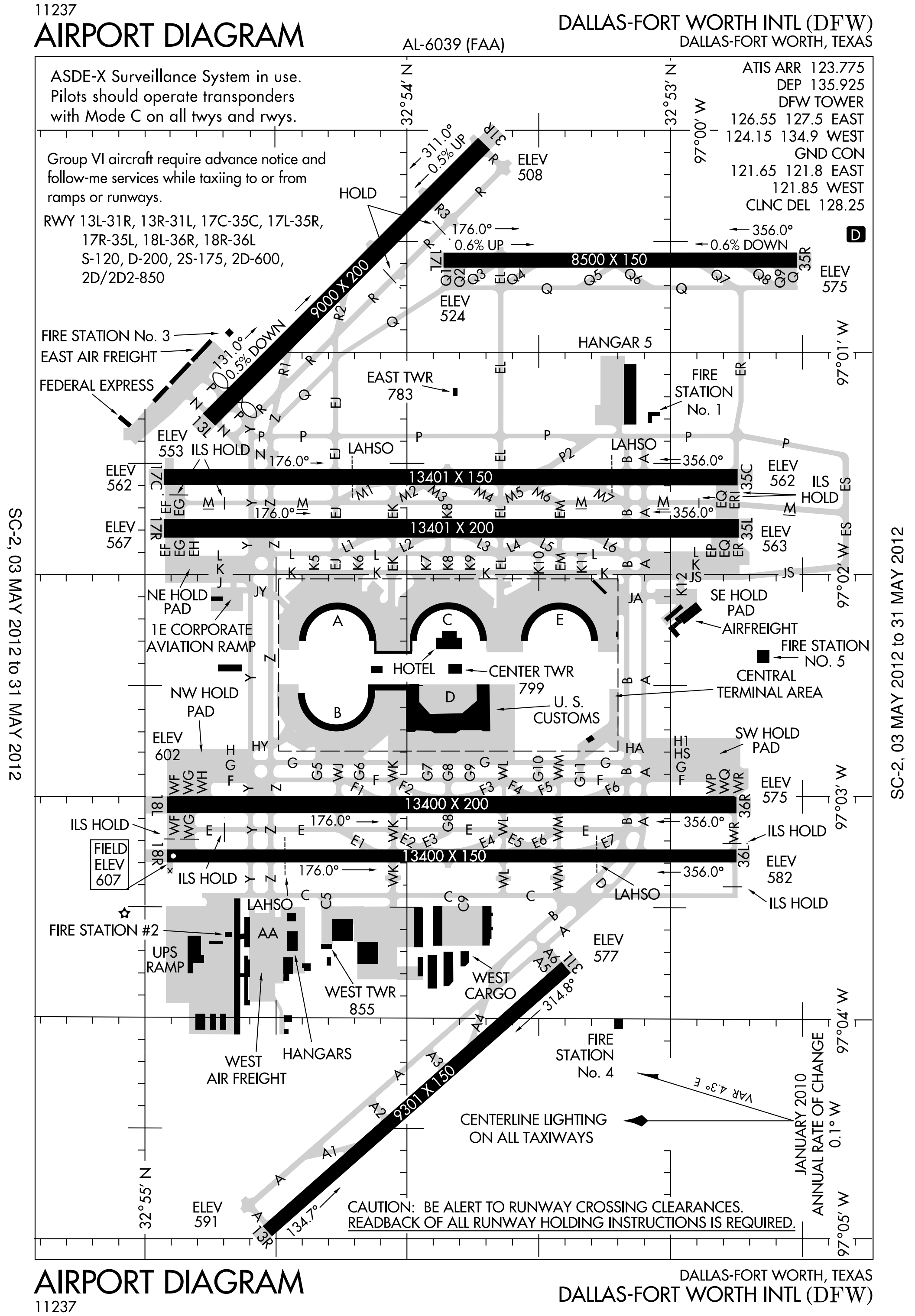

Figure C-1: DFW Airport Diagram.

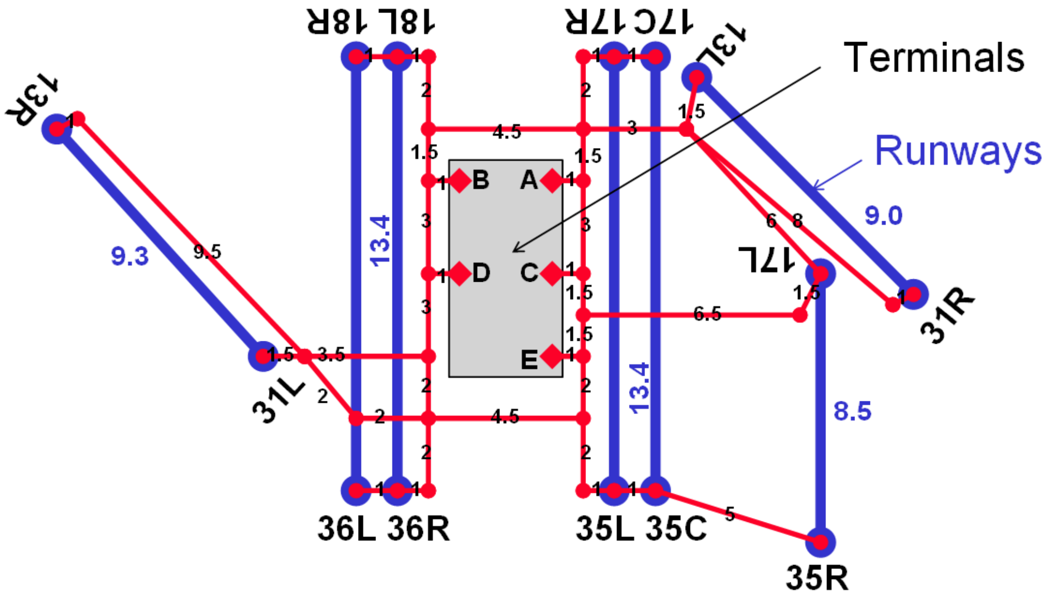

Figure C-2: Simplified DFW terminal area representation (distances in 1000s of feet).

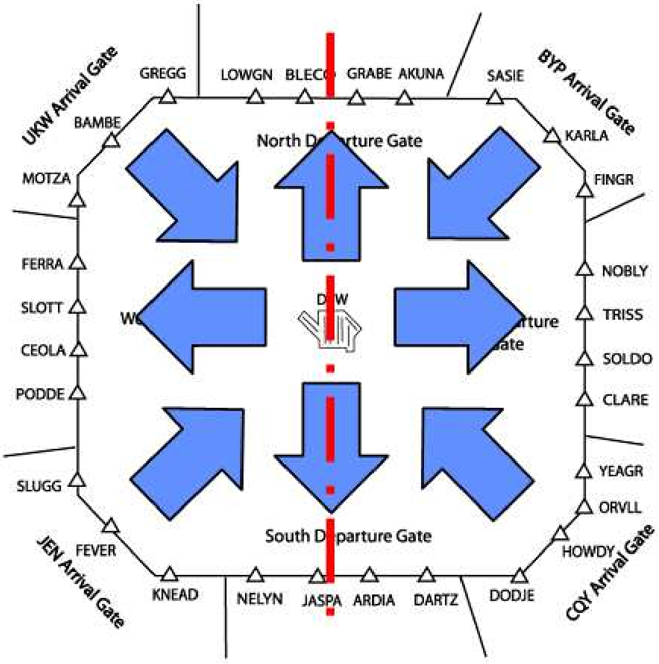

Figure C-3: Diagram of the DFW arrival and departure fixes.

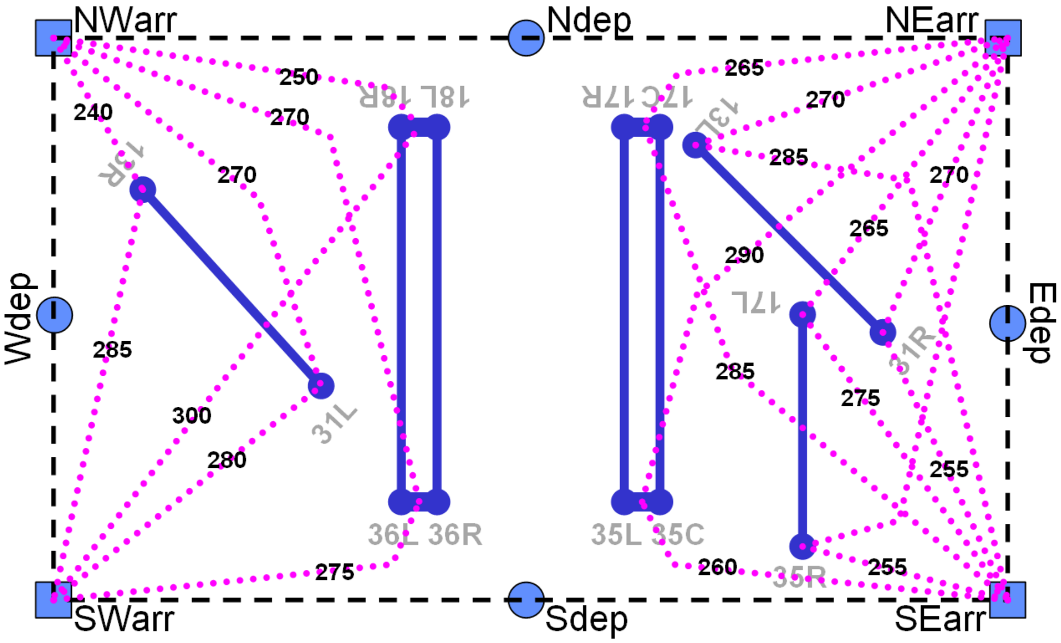

Figure C-4: Simplified DFW near-terminal arrival airspace representation (distances in 1000s of feet).

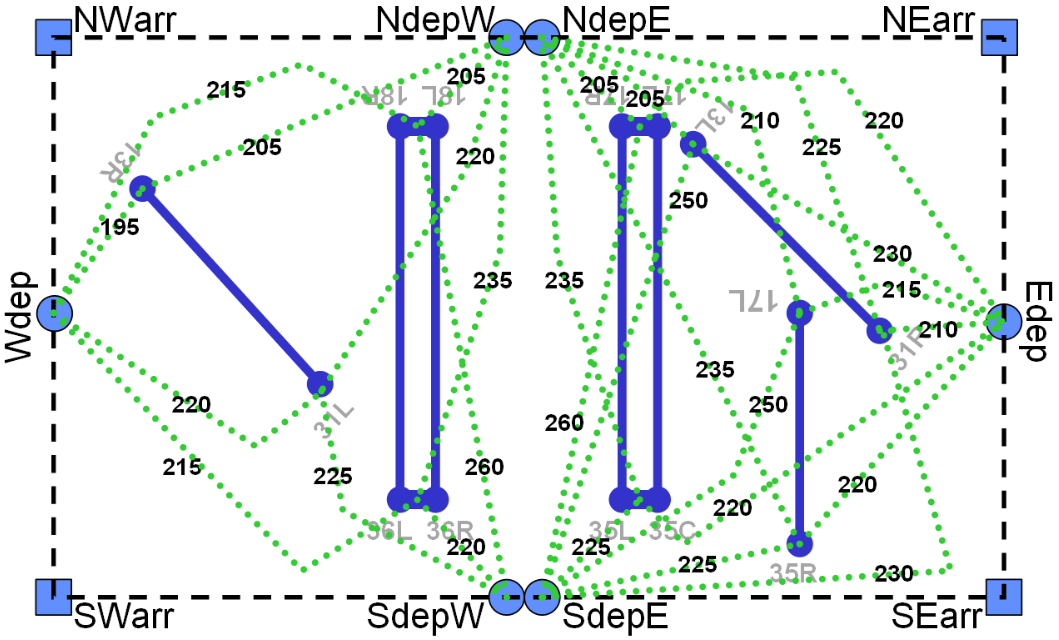

Figure C-5: Simplified DFW near-terminal departure airspace representation (distances in 1000s of feet).

Bibliography

[1] Ioannis Anagnostakis and John-Paul Clarke. Runway operations planning: A two-stage solution methodology. The 36th Hawaii International Conference on System Sciences, HI, 2003.

[2] H. Balakrishnan and B. Chandran. Algorithms for scheduling runway operations under constrained position shifting. *Operations Research*, 58(6), 2010.

[3] Michael Ball, Cynthia Barnhart, Martin Dresner, Mark Hansen, Kevin Neels, Amedeo Odoni, Everett Peterson, Lance Sherry, Antonio Trani, and Bo Zou. Total delay impact study – a comprehensive assessment of the costs and impacts of flight delay in the united states. Technical report, NEXTOR, November 2010. Accessed at http://www.isr.umd.edu/NEXTOR/pubs/TDI_Report_Final_11_03_10.pdf.

[4] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424, 2000.

[5] Dimitris Bertsimas and Michael Frankovich. Unified optimization of traffic flows through airports. Submitted to *Transportation Science*, June 2012.

[6] Dimitris Bertsimas, Michael Frankovich, and Amedeo Odoni. Optimal selection of airport runway configurations. *Operations Research*, 59(6):1407–1419, 2011.

[7] Dimitris Bertsimas, Guglielmo Lulli, and Amedeo Odoni. An integer optimization approach to large-scale air traffic flow management. *Operations Research*, 59(1):211–227, 2011.

[8] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[9] Dimitris Bertsimas and Sarah Stock-Patterson. The air traffic flow management problem with enroute capacities. *Operations Research*, 46(3):406–422, 1998.

[10] P. Bonnefoy. *Scalability of the Air Transportation System and Development of Multi-Airport Systems: A Worldwide Perspective*. PhD thesis, Engineering Systems Division, Massachusetts Institute of Technology, Cambridge, MA, 2008.

[11] P. Bonnefoy and R. J. Hansman. Emergence and impact of secondary airports in the united states. 6th USA/Europe Air Traffic Management R&D Seminar, Baltimore, MD, July 2005.

[12] P. Burgain. *On the control of airport departure processes.* PhD thesis, Georgia Institute of Technology, 2010.

[13] F. Carr. Stochastic modeling and control of airport surface traffic. Master's thesis, Massachusetts Institute of Technology, 2001.

[14] Sean X. Chen and Jun S. Liu. Statistical applications of the poisson-binomial and conditional bernoulli distributions. *Statistica Sinica*, 7:875–892, 1997.

[15] G.L. Clare and A.G. Richards. Optimization of taxiway routing and runway scheduling. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1000–1013, December 2011.

[16] R. de Neufville and A. Odoni. *Airport systems: Planning, Design, and Management.* McGraw-Hill, New York, November 2003.

[17] Roger G. Dear. The dynamic scheduling of aircraft in the near terminal area, research report r76-9. Technical report, Flight Transportation Laboratory, Massachussetts Insitute of Technology, Cambridge, MA, September 1976.

[18] FAA. National safety and operational criteria for runway use programs, November 1981. Accessed at http://www.faa.gov/documentLibrary/media/Order/8400-9.pdf.

[19] FAA. Long-range aerospace forecasts fiscal years 2020, 2025 and 2030, September 2007. Accessed at http://www.faa.gov/about/office_org/headquarters_offices/apl/aviation_forecasts/long-range_forecasts/media/long07.pdf.

[20] FAA. Dallas-Fort Worth International Airport diagram, May 2012. Accessed at http://www.faa.gov/airports/runway_safety/diagrams/index.cfm?print=go.

[21] E. R. Feron, R. J. Hansman, A. R. Odoni, R. B. Cots, B. Delcaire, W. D. Hall, H. R. Idris, A. Muharremoglu, and N. Pujet. The departure planner: A conceptual discussion. Technical report, Massachusetts Institute of Technology, 1997.

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

[23] E. P. Gilbo. Optimizing airport capacity utilization in air traffic flow management subject to constraints at arrival and departure fixes. *IEEE Transactions on Control Systems Technology*, 5(5):490–503, 1997.

[24] E. P. Gilbo and K. W. Howard. Collaborative optimization of airport arrival and departure traffic flow management strategies for CDM. 3rd USA/Europe Air Traffic Management R&D Seminar, Naples, Italy, June 2000.

[25] J-B. Gotteland, R. Deau, and N. Durand. Airport surface management and runways scheduling. 8th USA/Europe Air Traffic Management R&D Seminar, Napa, CA, 2009.

[26] Gautam Gupta, Waqar Malik, and Yoon C Jung. Effect of uncertainty on deterministic runway scheduling. 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Virginia Beach, VA, September 2011.

[27] William D. Hall. *Efficient Capacity Allocation in a Collaborative Air Transportation System*. PhD thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, 1999.

[28] S. Kellner and D. Kösters. Contribution to the ATMAP project. Working paper, Institute of Transport Science, RWTH Aachen University, November 2008.

[29] Waqar Malik, Gautam Gupta, and Yoon Jung. Managing departure aircraft release for efficient airport surface operations. AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, Canada, 2010.

[30] Ángel G. Marín. Airport management: Taxi planning. *Annals of Operations Research*, 143:191–202, 2006.

[31] N90 TRACON - LaGuardia Standard Operating Procedures, November 2008.

[32] Diana Michalek Pfeil. *Optimization of Airport Terminal-Area Air Traffic Operations under Uncertain Weather Conditions*. PhD thesis, Operations Research Center, Massachusetts Institute of Technology, 2011.

[33] Plymouth state weather center WXP surface text data generator, 2010. Accessed at http://vortex.plymouth.edu/sa_parse-u.html.

[34] Harilaos N. Psaraftis. A dynamic programming approach for sequencing groups of identical jobs. *Operations Research*, 28(6):1347–1359, 1980.

[35] N. Pujet, B. Delcaire, and E. Feron. Input-output modeling and control of the departure process of congested airports. AIAA Guidance, Navigation, and Control Conference and Exhibit, pages 1835–1852, Portland, OR, 1999.

[36] Varun Ramanujam and Hamsa Balakrishnan. Estimation of arrival-departure capacity tradeoffs in multi-airport systems. Proceedings of the 48th IEEE Conference on Decision and Control, December 2009.

[37] Varun Ramanujam and Hamsa Balakrishnan. Estimation of maximum-likelihood discrete-choice models of the runway configuration selection process. Proceedings of the American Control Conference, June 2011.

[38] S. Rathinam, J. Montoya, and Y. Jung. An optimization model for reducing aircraft taxi times at the dallas fort worth international airport. 26th International Congress of the Aeronautical Sciences (ICAS), pages 14–19, 2008.

[39] Alexander Shapiro and Andy Philpott. A tutorial on stochastic programming, March 2007. Accessed at http://www.epoc.org.nz/publications.html.

[40] I. Simaiakis, H. Khadilkar, H. Balakrishnan, T. G. Reynolds, R. J. Hansman, B. Reilly, and S. Urlass. Demonstration of reduced airport congestion through pushback rate control. Technical Report ICAT-2011-2, MIT International Center for Air Transportation (ICAT), January 2011.

[41] Gustaf Sölveling, Senay Solak, John-Paul Clarke, and Ellis Johnson. Runway operations optimization in the presence of uncertainties. *Journal of Guidance, Control, and Dynamics*, 34(5), 2011.

[42] M. A. Stamatopoulos, K. G. Zografos, and A. R. Odoni. A decision support system for airport strategic planning. *Transportation Research Part C*, 12:91–117, 2004.

[43] W. J. Swedish. Upgraded FAA airfield capacity model. Rep. FAA-EM-81-1, Vol. 1 and 2, The MITRE Corporation, McLean, VA, 1981.

[44] Dionyssios A. Trivizas. *Parallel parametric combinatorial search – its application to runway scheduling*. PhD thesis, Flight Transportation Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1987.

[45] John N. Tsitsiklis. Special cases of traveling salesman and repairman problems with time windows. *Networks*, 22:263–282, 1992.