Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)

Air Traffic Flow Management at Airports: A Unified Optimization Approach

Dimitris Bertsimas, Michael Frankovich
Aviation Edge, LLC
Belmont, MA, USA

Dimitris Bertsimas
Operations Research Center
Massachusetts Institute of Technology
Cambridge, MA, USA

***Abstract*—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 in which mode they operate; b) assigning flights to runways and determining the sequence in which flights are processed (i.e., when they take off or land); c) determining the gate-holding duration of departures; and d) routing flights to their assigned runway and onwards within the terminal area and the near-terminal airspace. The key contribution of this paper is the modeling of these problems, which until present have been studied mainly in isolation, under a framework which is both unified and tractable. This allows the possibility of obtaining system-optimal solutions in a practical amount of time. Furthermore, the approach is implemented on historic datasets from both Boston Logan International (BOS) and Dallas/Fort Worth (DFW) airports. Computational experience indicates that significant improvements can be achieved from this optimization, and that computational tractability is such that real-world implementation is possible.**

Keywords-air traffic flow management; integer programming applications; reduced emissions, ground delay; airborne delay; runway separation; gate holding; departure metering; configuration selection; taxi routing.

I. 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 ([13]).

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, and when previous studies have focused on optimizing operations at airports, they have largely 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 work.

This airport-centric approach to optimizing national air traffic 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 paper can be used to determine AARs and thus affect air traffic on the network level through the use of GDPs.

In this paper, we seek in particular 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, if any;

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

A. Contributions of the Paper

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. We feel that 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 sub-optimal, or indeed infeasible.

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.

II. BACKGROUND

There has been much work on these and related subproblems within the aviation and optimization communities, but this work has focused mainly on a single subproblem at a time in isolation: Gilbo [16] presented an integer optimization model for the arrival/departure runway balancing (ADRB) problem by modeling runway capacity using a runway configuration capacity envelope (RCCE). Bertsimas et al. [5] then solved the airport runway configuration management (RCM) problem (a) above, and the (ADRB) problem in a single optimization model, as well as proposing an extension to the case of airports in a metroplex with shared airspace. This work is more strategic in nature to that of this paper 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 paper, 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.

The sequencing problem of (b) above 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 [26], 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 [26]) as well as in this application. Notably, [12], [20], [25], 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, [24] proposed a stochastic optimization approach to the runway scheduling problem.

There is also a substantial body of work on airport surface management and the gate-holding of departures which relates to subproblems (c) and (d) above. 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 [14], [21], [9], and [8]. Notably, [23] 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 in queue at the runways.

Marín [19], [22], and [18] proposed approaches to the optimization of aircraft taxi routes, while [17], and more recently [10], merged the sequencing problem with the taxiing problem, recognizing as does this paper the important interdependence between the two problems. However, their work ignores the important and complicating matter of runway configuration optimization.

III. OVERVIEW OF THE MODEL

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; as well as for every arrival: i) a time at arrival fix (which may imply a speed control policy before reaching the fix); ii) a runway assignment and gate assignment; iii) 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 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.

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 Phase Two 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 IV will show there to be almost no loss of optimality in the real-world instances to which we apply our methodology.

A. Phase One

1) Data

First we detail the data requirements of the Phase One optimization problem. We consider a time horizon T = $\{1, ..., T\}$ 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 F, with each flight having a weight class w (heavy, large, small, or Boeing-757) and an orientation o (arrival or departure). The pair $i = (w, o)$ will be referred to as a flight type, belonging to the set of flight types C (the index i will always refer to a flight type in Phase One). 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 K. Each configuration k is described by a set of pairs $\{(r_1, m_1), ..., (r_N, m_N)\}$, 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:

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

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

C_A, C_D = the set of flight types whose orientation is arrival, departure, respectively;

$F = F_A \cup F_D = \cup_i F_i$ = the set of flights (and of arrivals, departures, flights of each type i);

R = the set of runways, each of which includes a single, fixed, direction of operation;

R_f (R_i) = the set of runways which is feasible for flight f (or for some flight of type i). The feasibility of a given runway for a given flight depends on several factors, notably aircraft type and runway dimensions;

V = the set of pairs of runways $\{(r_1, r_2),\ldots, (r_{2N-1}, r_{2N})\}$ where separation must be enforced between them, such as runways which physically cross each other;

K = the set of runway configurations, each of which is a set of pairs $k = \{(r_1, m_1),\ldots,(r_N, m_N)\}$, 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;

R_k = the set of runways used by configuration k;

I_{rk} = the set of flight types that can use runway r under configuration k;

U_t = those runways which cannot be used at time t due, for example, to strong crosswinds to tailwinds;

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

$\underline{T}^f_{o(f)}$ = the release time of flight f from its origin $o(f)$ (gate or arrival fix) into the system;

l^i_r = the number of time intervals constituting the runway occupancy time of flights of type i at runway r;

s^r_{ij} = the minimum number of time intervals of separation required between aircraft when an aircraft of type j follows an aircraft of type i at runway r. We refer the reader to [11] for more details, but we note that this is always at least equal to l^i_r, the runway occupancy time of the first aircraft;

$s^{(r, r')}_{ij}$ = the minimum number of time intervals of separation required at crossing runways when an aircraft of type j scheduled at runway r' follows an aircraft of type i scheduled at runway r, if $(r, r') \in$ V;

β^A_A, β^A_D = constants weighting the delay cost in the air for arrivals and departures, respectively, with $\beta^A_A > \beta^A_D$;

β^G_A, β^G_D = 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^f_r = 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.

2) Decision Variables

We define the following binary decision variables for our model:

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

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

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

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

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 $O(|\mathrm{C}||\mathrm{R}||\mathrm{T}| + |\mathrm{F}||\mathrm{R}|)$, rather than $O(|\mathrm{F}||\mathrm{R}||\mathrm{T}|)$, and the number of constraints to $O(|\mathrm{C}|^2|\mathrm{R}||\mathrm{T}|)$, rather than $O(|\mathrm{F}|^2|\mathrm{R}||\mathrm{T}|)$. Indeed, this modeling technique may be applied for general bounded-TSP type problems.

3) Objective Function

Our objective is to minimize a weighted sum of delays:

$$\min \Psi = \sum_{i \in \mathrm{C}}\left(\beta_D^G \mathbb{I}_{\{i \in \mathrm{C}_D\}} + \beta_A^A \mathbb{I}_{\{i \in \mathrm{C}_A\}}\right) \sum_{r \in \mathrm{R}_i} \sum_{t \in \mathrm{T}} t\psi_{rt}^i - \sum_{f \in \mathrm{F}}\left(\beta_D^G \mathbb{I}_{\{f \in \mathrm{F}_D\}} + \beta_A^A \mathbb{I}_{\{f \in \mathrm{F}_A\}}\right) \underline{T}_{o(f)}^f + \sum_{f \in \mathrm{F}} \sum_{r \in \mathrm{R}_f}\left(\beta_D^A \mathbb{I}_{\{f \in \mathrm{F}_D\}} + \beta_A^G \mathbb{I}_{\{f \in \mathrm{F}_A\}}\right) d_r^f \varphi_r^f - (\beta_D^G - \beta_G)\left(\sum_{i \in \mathrm{C}_D} \sum_{r \in \mathrm{R}_i} \sum_{t \in \mathrm{T}} t\psi_{rt}^i - \sum_{f \in \mathrm{F}_D} \sum_{r \in \mathrm{R}_f} \underline{T}_r^f \varphi_r^f\right) + K \sum_{t \in \mathrm{T}} \chi_t.$$

This can be summarized in words as a summation over all flights of the following terms: (weighted time from first time period until touchdown/takeoff) – (weighted time from first time period until start time) + (weighted time from touchdown/takeoff to destination) – (weighted gate-holding duration) + (configuration change penalty).

4) Model Formulation

The constraints of the model, which we call P1, are as follows:

$$\sum_{k \in \mathrm{K}} \omega_{kt} = 1, \forall t \in \mathrm{T} \quad (1)$$

$$\psi_{rt}^i = 0, \forall i \in \mathrm{C}, r \in \mathrm{U}_t, t \in \mathrm{T} \quad (2)$$

Constraints (1) require exactly one configuration to be used at any time, while Constraints (2) 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). In addition, it enables us to add an extra class of valid inequalities to strengthen the model.

$$\psi_{r,t-h}^i + \psi_{rt}^j \leq 1, \forall i,j \in \mathrm{C}, r \in \mathrm{R}_i \cap \mathrm{R}_j, h \in \left\{1, \ldots, \min\left\{s_{ij}^r - 1, t-1\right\}\right\}, t \in \mathrm{T}\backslash\{1\} \quad (3)$$

Constraints (3) can generally be referred to as the separation constraints, which state that if we process a flight of type *i*, then we must wait at least s^r_{ij} time periods before processing a flight of type *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.

$$\psi_{r,t-h}^i + \psi_{r't}^j \leq 1, \forall i,j \in \mathrm{C}, (r,r') \in \left(\mathrm{R}_i \times \mathrm{R}_j\right) \cap \mathrm{V}, h \in \left\{0, \ldots, \min\left\{s_{ij}^{(r,r')} - 1, t-1\right\}\right\}, t \in \mathrm{T} \quad (4)$$

Constraints (4) enforce a similar separation requirement when we have a pair of runways *(r, r')* which cross over each other. We also need to consider the case of close parallel runways, where operations on each runway are not independent. In this paper we take the (only slightly) conservative approach of modeling this situation as though there is only a single runway, although we could alternatively include appropriate additional constraints to the model.

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, as in [11]. When flights use different runways and/or different fixes, the relevant separation requirements will be enforced in our second stage problem. Due to the nature of runway configurations, where a general flow in one direction is preserved, these second-stage constraints will not significantly impact the overall optimality of our two-phase approach.

$$\sum_{i \in \mathrm{C}: r \in \mathrm{R}_i} \psi_{rt}^i \leq 1, \forall r \in \mathrm{R}, t \in \mathrm{T} \quad (5)$$

$$\psi_{rt}^i + \omega_{kt} \leq 1, \forall t \in \mathrm{T}, k \in \mathrm{K}, r \in \mathrm{R}_k, i \in \bar{\mathrm{I}}_{rk} : r \in \mathrm{R}_i \quad (6)$$

$$\psi_{rt}^i - \sum_{k \in \mathrm{K}: r \in \mathrm{R}_k, i \in \mathrm{I}_{rk}} \omega_{k,t+h} \leq 0, \forall i \in \mathrm{C}, r \in \mathrm{R}_i, h \in \{0, \ldots, min\{l_i^r - 1, T - t\}\}, t \in \mathrm{T} \quad (7)$$

We now remind the reader that the definition of $\boldsymbol{\psi}$ is such that $\psi^i_{rt} = 1$ if, and only if, a flight of type *i* arrives at runway *r* at time *t*, and hence such a flight might (and in general, will) actually occupy the runway for a longer period of time, even though this is not directly tracked by our decision variables $\boldsymbol{\psi}$. Then, Constraints (5) state that only one flight may arrive at each runway at any given time. This set of constraints, along with Constraints (3) above, enforce the capacity of each runway to be one at all times (recall $s^r_{ij} \geq l^i_r$). Constraints (6) disallow the use of runway *r* for flights of type *i* if such use is not allowed under configuration *k*. Constraints (7) state that if we process a flight of type *i* at a given runway *r*, then that runway must remain open for at least l^i_r time periods, corresponding to the runway occupancy time of flights of type *i*.

$$\sum_{r \in \mathrm{R}_f} \varphi_r^f = 1, \forall f \in \mathrm{F} \quad (8)$$

$$\sum_{f \in \mathrm{F}_i : r \in \mathrm{R}_f, t \geq \overline{T}_r^f - l_i^r + 1} \varphi_r^f \leq \sum_{\tau=1}^{t} \psi_{r\tau}^i \leq \sum_{f \in \mathrm{F}_i : r \in \mathrm{R}_f, t \geq \underline{T}_r^f} \varphi_r^f, \forall i \in \mathrm{C}, r \in \mathrm{R}_i, t \in \mathrm{T} \quad (9)$$

Constraints (8) state that the every flight must be assigned to some runway. Then, Constraints (9) require each flight f to be processed at one of its feasible runways r after its earliest possible touchdown/takeoff time. 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 i assigned to runway r by time t must fall within this range, for every $t \in \mathrm{T}$. These are the only constraints that link the $\boldsymbol{\psi}$ variables with the $\boldsymbol{\varphi}$ variables.

$$\chi_t - \omega_{kt} + \omega_{k,t-1} \geq 0, \forall k \in \mathrm{K}, t \in \mathrm{T} \backslash \{1\} \quad (10)$$

Finally, Constraints (10) enforce $\chi_t = 1$ if a change of configuration occurs at time t, which happens if, and only if there exist a k such that $\omega_{kt} = 1$ and $\omega_{k,t-1} = 0$. 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 $\boldsymbol{\chi}$.

5) Remarks on the Model

- A helpful way to think about this model is to first suppose that the taxiway/near-terminal area airspace network has infinite capacity. In this case, all flights can travel along their shortest paths without obstruction and hence arrive at their assigned runway within their time-window specified in the input data, and in particular at their assigned time (as discussed below, this can be derived from $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$). Then, P1 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^f_{rt}, 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 were to grow, especially due to the number of constraints required to enforce minimum between-flight separation regulations. 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 $\psi^i_{rt} = 1$ if a flight of type 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 the time-window constraints (9) that we link these two sets of variables. Indeed, finding flight type slots and then allocating specific flights to these slots has also been proposed by [1] and [24]. Inspection of constraints (9) reveals that there is always at least one sequence of flights corresponding to a solution of P1, and that such a sequence can be trivially obtained from the solution, assuming flight deadlines are not strict.

B. Phase Two

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 I and in this second phase we obtain the solution to subproblems (c) and (d). In more detail, we present a binary optimization problem, P2, which takes the solution from P1 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 P2 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 IV, we shall support this statement.

1) Data

In order to model the routing subproblem, the airport network is represented by a directed graph with nodes belonging to the set S, where each node represents a section of taxiway, a runway, an airspace route, a gate, or a fix. A full list of relevant sets and parameters, building on those above, is given below:

S (S_f) = the set of nodes in the airport network (feasible for flight f);

L^f_i = the set of nodes which are successors of node i for flight f;

P^f_i = the set of nodes which are predecessors of node i for flight f;

E_f = the set of possible end nodes of flight f;

$\mathrm{T}_f = \{T_f, T_f+1, \ldots, T_f\}$ = 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;

c_f ($\in$ C) = the type of flight f;

o_f = initial node of flight f;

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

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

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 P1 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.

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

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

$\mathrm{H}_{(r,r')} = \{(f_1, f_2), (f_2, f_3), \ldots, (f_{n-1}, f_n)\}$ = 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;

$\mathrm{W}_i = \{(f_1, f_2), (f_2, f_3), \ldots, (f_{n-1}, f_n)\}$ = 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 $f \rightarrow g$ "within close proximity" of each other, and require $s^i_{c(f),c(g)}$ time intervals of separation there.

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 1 gives an example of an element of Q. This set will be used to ensure that configuration requirements are respected, since they are not modeled explicitly in the model P2 below.

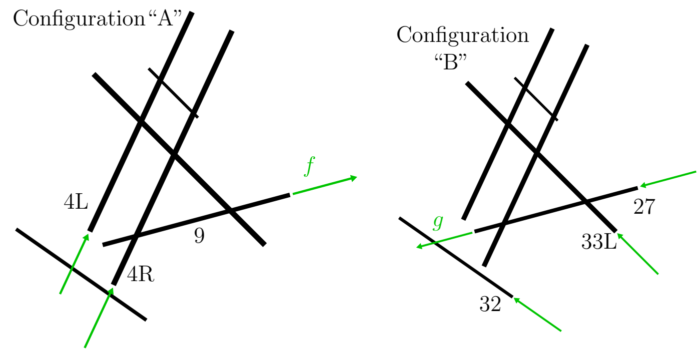

FIGURE 1: *Example illustrating an element (f, g) belonging to the set Q at BOS. The arrows indicate the direction and mode of traffic as dictated by the configuration in use. Suppose in the solution to P1 we have: i) configuration A is used first, then configuration B, ii) flight f is assigned to runway 9, and flight g to runway 27. Since runway 27 is not used in configuration A, we have (f, g) ∈ Q.*

2) *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 [7], which will lead to nice properties in the model formulation.

3) *Objective Function*

The objective function, which we omit for brevity, is to minimize the same quantity as the P1 objective function – the weighted sum over the total time it takes for each flight to go through the system, between the gates and the arrival/departure fixes.

4) *Model Formulation*

The following constraints complete the binary optimization problem P2, which routes flights to achieve the schedule of assigned runways and assigned flight sequences at each runway which were found in P1. The model is based on the models of [6] and [7], which were presented to solve the network ATFM problem with and without re-routing, respectively.

$$x^f_{jt} - \left(z^f_{jt} - \sum_{i \in \mathrm{L}^f_j : t \geq \underline{T}^f_i} z^f_{it} \right) = 1, \forall f \in \mathrm{F}, \mathrm{j} \in \mathrm{S}_f, t \in \mathrm{T}^f_j \quad (11)$$

$$\sum_{f \in \mathrm{F}: j \in \mathrm{S}_f, t \in \mathrm{T}^f_j} x^f_{jt} \leq u_{jt}, \forall \mathrm{j} \in \mathrm{S} \backslash \mathrm{R}, t \in \mathrm{T} \quad (12)$$

Constraints (11) link the $\boldsymbol{x}$ variables with the $\boldsymbol{z}$ variables, with x^f_{jt} 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 (12) 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 P1).

$$z_{jt}^{f} - \sum_{i \in \mathrm{P}_j^f : t - l_i^f \geq \underline{T}_i^f} z_{i,t-l_i^f}^{f} \leq 0, \forall f \in \mathrm{F}, \mathrm{j} \in \mathrm{S}_f \backslash \{o_f\}, t \in \mathrm{T}_j^f \quad (13)$$

$$z_{i\overline{T}_i^f}^{f} - \sum_{j \in \mathrm{L}_i^f} z_{j\overline{T}_j^f}^{f} \leq 0, \forall f \in \mathrm{F}, \mathrm{i} \in \mathrm{S}_f \backslash \mathrm{E}_f \quad (14)$$

$$\sum_{j \in \mathrm{E}_f} z_{j\overline{T}_j^f}^{f} = 1, \forall f \in \mathrm{F} \quad (15)$$

Constraints (13) 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 (14) 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 (15) state that the flight must reach one of its feasible destinations.

$$\sum_{j \in \mathrm{L}_i^f} z_{j\overline{T}_j^f}^{f} \leq 1, \forall f \in \mathrm{F}, \mathrm{i} \in \mathrm{S}_f \quad (16)$$

$$z_{j,t-1}^{f} - z_{jt}^{f} \leq 0, \forall f \in \mathrm{F}, \mathrm{j} \in \mathrm{S}_f, t \in \mathrm{T}_j^f \backslash \{\underline{T}_j^f\} \quad (17)$$

$$z_{o_f \underline{T}_{o_f}}^{f} = 1, \forall f \in \mathrm{F} \quad (18)$$

Constraints (16) 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 (17) enforce monotonicity on the *z* variables, owing to their definition. Constraints (18) initialize each flight at its origin.

$$z_{r_f \overline{T}_{r_f}}^{f} = 1, \forall f \in \mathrm{F} \quad (19)$$

$$z_{r,t+s_{c(g),c(f)}^{r}}^{f} - z_{rt}^{g} \leq 0, \forall r \in \mathrm{R}, (\mathrm{g},\mathrm{f}) \in \mathrm{H}_r, \forall t \in \mathrm{T}: t \geq \underline{T}_r^g \text{ and } \mathrm{t} + \mathrm{s}_{c(g),c(f)}^{r} \in \mathrm{T}_r^f \quad (20)$$

$$z_{r',t+s_{c(g),c(f)}^{(r,r')}}^{f} - z_{rt}^{g} \leq 0, \forall (r,r') \in \mathrm{V}, (\mathrm{g},\mathrm{f}) \in \mathrm{H}_{(r,r')}, \forall t \in \mathrm{T}: t \geq \underline{T}_r^g \text{ and } \mathrm{t} + \mathrm{s}_{c(g),c(f)}^{(r,r')} \in \mathrm{T}_{r'}^f \quad (21)$$

Constraints (19) force a flight to use its assigned runway from P1. Constraints (20) state that flights must be processed at each runway in the order determined from P1, and be separated by at least the minimum separation time, while Constraints (21) enforce these same ordering and separation requirements for the pairs of flights scheduled on intersecting/closely-spaced parallel runways.

$$z_{i,t+s_{c(f),c(g)}^{i}}^{g} - z_{it}^{f} \leq 0, \forall (f,g) \in \mathrm{W}_i, i \in \mathrm{S}, \forall t \in \mathrm{T}_i^f : \mathrm{t} + \mathrm{s}_{c(f),c(g)}^{i} \in \mathrm{T}_i^g \quad (22)$$

Constraints (22) 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 (20), (21) and (22) 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 H_r, $\mathrm{H}_{(r,r')}$ and W. Second, due to the form of the constraints, which state that one set of the "by" variables *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.

$$z_{r_f,t+l_{r_g}^g}^{f} - z_{r_g,t}^{g} \leq 0, \forall (\mathrm{g},\mathrm{f}) \in \mathrm{Q}, \forall t \in \mathrm{T}: t \geq \underline{T}_{r_g}^g \text{ and } \mathrm{t} + \mathrm{l}_{r_g}^g \in \mathrm{T}_{r_f}^f \quad (23)$$

$$x_{rt}^{f} = 0, \forall f \in \mathrm{F}, r \in \mathrm{U}_t, t \in \mathrm{T}_r^f \quad (24)$$

Constraints (23) ensure that the configuration requirements are respected by ensuring that we process all pairs of flights in the set Q in the specified order. Note that we have defined the set Q to be as small as possible while still preventing the operation of illegal configurations, expanding the feasible space of P2. Constraints (24) 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).

$$z_{it}^{f} = z_{i\overline{T}_i^f}^{f}, \forall f \in \mathrm{F}, i \in \mathrm{S}_f, t \in \left\{\overline{T}_i^f + 1, \ldots, T\right\} \quad (25)$$

Finally, Constraints (25) extend the *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 (11) correctly define the variables ***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_{it}^f.

IV. Computational Experience

In this section we present 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, being 11/02/2009 at DFW and 9/28/2010 at BOS.

All experiments were performed using the software package GUROBI 5.0 on a computer with an Intel® Core™ 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 P1 and P2.

1) Model Validation and Computational Tractability

Tables 1 and 2 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-stage approach, and ii) to demonstrate computational tractability on real-world instances.

The first thing to note is that the guarantee of optimality (obtained by comparing solutions from Phase One and Phase Two, this value is an *upper bound* on the optimality gap of our feasible solution) is always very good – aside from the greatest difference of 2.1%, the others are at most 1.1%, and in the former case 1% of the gap is due to P2 not being solved to optimality. 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 instances. However, in some instances P2 can take significantly longer than the median. In these cases, we typically have a good solution much earlier that at termination.

TABLE 1 Computational Tractability and a Bound on the Optimality Gap Using Data from DFW on 11/2/2009

Flights	Optimality Bound	Computational Times (s)		
		P1	***P2***	***Total***
155	2.1	120	1286	1430
175	1.1	372	1071	1465
153	0.6	64	129	211
155	0.7	75	284	379
168	0.5	340	187	546
171	0.7	299	284	600
159	0.9	252	533	806
153	0.6	241	205	463

TABLE 2 Computational Tractability and a Bound on the Optimality Gap Using Data from BOS on 9/28/2010

Flights	Optimality Bound	Computational Times (s)		
		P1	***P2***	***Total***
90	0.2	252	147	418
91	0.4	233	142	388
80	0.1	143	16	168
63	0.6	93	52	161
63	0.3	198	15	229
71	1.3	246	1200	1457
59	0.5	255	59	325
63	0.3	161	17	187

2) Benefits Assessment

Above we have demonstrated 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 and 4 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 means 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 β place a higher emphasis on reducing 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.

TABLE 3 Comparison of Optimized and Historic Surface Times at DFW on 11/2/2009

Optimized Surface Times (min./flight)				Historic Surface Times (min./flight)		
Dep. G.H.	***Dep.***	***Arr.***	***Avg.***	***Dep.***	***Arr.***	***Avg.***
1.8	9.5	10.8	10.2	13	8.9	10.7
2.2	9.2	10.7	9.9	12.8	9.3	11.2
1	9.5	10.6	10	13.5	9.2	11.6
1	9.6	11.1	10.4	13.5	9	11
0.9	9.3	11.8	10.5	13	10.1	11.6
2.1	10	11.4	10.7	13.6	8.9	11.3
1.3	9.1	10.8	10	13.9	9.1	11.4
0.7	9	11.1	10.1	13.4	9.6	11.2

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. However, this is due to the relatively low weight we place on arrival taxi times – the model sacrifices these slightly for the sake of reduced air delays and departure taxi times. 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.

TABLE 4 Comparison of Optimized and Historic Surface Times at BOS on 9/28/2010

Optimized Surface Times (min./flight)				Historic Surface Times (min./flight)		
Dep. G.H.	***Dep.***	***Arr.***	***Avg.***	***Dep.***	***Arr.***	***Avg.***
1.6	13.7	4.4	9.3	18.7	5.2	12.4
0.8	14	4.5	9.6	16.7	6.1	11.6
2.2	16.4	5	12.2	17.6	5.9	13.1
0.5	16.6	4.9	10.3	18.2	6.5	12.2
0.5	16	5	10.1	18.5	4.4	10.9

1.3	11.6	8.2	10.1	16.6	5.1	11.4
2.6	13.8	4.6	8.5	14.7	5.8	9.9
0.9	16.2	4.9	10.5	17.1	6.6	11.3

3) Summary of Findings

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

Conclusion

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.

We note two important points related to any potential implementation of the methodology outlined in this paper. Firstly, we have assumed all data inputs to be deterministically known, keeping the focus of the work away from the dynamic and uncertain nature of the real-world environment in which such a methodology could be used. Secondly, 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. Both of these represent key areas for future research. Nevertheless, the methodology we have developed represents a significant step towards improving ATFM at airports.

Acknowledgment

This work was supported by MIT Lincoln Laboratory. We would like to thank James Kuchar, Bill Moser, Tom Reynolds, Ngaire Underhill and Mark Weber of MIT Lincoln Laboratory for their assistance with data. We would also like to thank Mr Ross Anderson and Dr Shubham Gupta for helpful discussions.

References

[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] Dimitris Bertsimas and Michael Frankovich. Unified optimization of traffic flows through airports. Unpublished, 2012.

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

[6] 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.

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

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

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

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

[11] R. de Neufville and A. Odoni. Airport systems: Planning, Design, and Manage- ment. McGraw-Hill, New York, November 2003.

[12] 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.

[13] 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.

[14] 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.

[15] Michael J. Frankovich. Air Traffic Flow Management at Airports: A Unified Optimization Approach. PhD thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, 2012.

[16] 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.

[17] 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.

[18] 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.

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

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

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

AUTHOR BIOGRAPHY

Dimitris J. Bertsimas is the Boeing Leaders for Global Operations Professor of Management at Massachusetts Institute of Technology (MIT), the codirector of the Operations Research Center at MIT, and a co-founder of Aviation Edge, LLC. He is also a member of the National Academy of Engineering. His research interests include discrete, robust, and stochastic optimization and their applications.

Michael J. Frankovich received his PhD in Operations Research from the Massachusetts Institute of Technology in 2012. He is a co-founder of Aviation Edge, LLC, and is interested in optimization and its applications in transportation, notably in the airline industry.