Unified Optimization of Traffic Flows Through Airports

Dimitris Bertsimas
Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, dbertsim@mit.edu

Michael Frankovich
Aviation Edge, LLC, Belmont, Massachusetts 02478, mfrankovich@aviation-edge.com

We present a novel integer optimization approach to optimize, in a tractable and unified manner, the airport operations optimization problem. This includes solving the entirety of key air traffic flow management 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 onward 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 in isolation, under a framework that 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 data sets from both Boston Logan International and Dallas/Fort Worth International airports. Computational experience indicates that significant reductions in delays, fuel use, and emissions can be achieved from this optimization and that computational tractability is such that real-world instances can be solved within five to 10 minutes.

Keywords: air traffic flow management; integer programming applications

History: Received: June 2012; revisions received: February 2014, November 2014; accepted: November 2014.
Published online in Articles in Advance August 17, 2015.

1. Introduction

In 2007, the cost of delays to U.S. domestic flights on major airlines was estimated to be $8.3 billion, and the cost to passengers $16.7 billion (Ball et al. 2010). Reducing these 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 but also to improve the experience for passengers. Furthermore, addressing these delays is important because 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 (FAA 2007).

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 among experts in the airline industry that infrastructure development alone will not be enough to limit significant increases in delays above current levels (Ball et al. 2010). As a result, there is a growing need to incorporate optimization into the air traffic flow management (ATFM) to minimize these delays. With reduced delays also come reductions in fuel consumption and emissions as well as improved safety management.

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 that 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 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, along with having implications at the airport being optimized itself, the work of this paper can be used to determine AARs (since it can accurately model an airport’s operations and hence capture its throughput capacity under different scenarios) 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, and the time at which arrivals should reach their arrival fix;

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

1.1. Literature Review

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 (1997) 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, Frankovich, and Odoni (2011) then solved the airport runway configuration management (RCM) problem (a) and the ADRB problem in a single optimization model as well as proposing an extension to the case of airports in a metropolis with shared airspace. That work is higher level 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 because of 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 toward determining a more precise, and time-varying, maximal throughput.

The sequencing problem of (b) 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, which is 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 takeoff/landing is safe to go ahead. In particular, the TRP problem here is a special case (B-TRPTW, to use the notation of Tsitsiklis 1992), 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 Tsitsiklis 1992) as well as in this application. Notably, Dear (1976), Psarafitis (1980), Trivizas (1987), and Balakrishnan and Chandran (2010) developed approaches that took advantage of the fairness principle that the optimal sequence should not differ too much from the first-come-first-served sequence. Recently, Solveling et al. (2011) proposed a stochastic optimization approach to the runway scheduling problem.

There is also a substantial body of work on the gate-holding of departures and delaying of arrivals, which relates to subproblem (c). One objective here is to hold departures at the gate, with engines off, for as long as possible without delaying their takeoff. In other words, delays in queue at runways or elsewhere in 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, Feron et al. (1997); Pujet, Delcaire, and Feron (1999); Carr (2001); and Burgain (2010). Notably, Simiakis et al. (2011) implemented a simple but effective “N-control” policy at Boston Logan International 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 the same way that departures can be held at the gate, arrivals can also be speed-controlled (or ground-held at their origin, depending on the time horizon considered) to delay their arrival at their arrival fix. Lowther (2008) developed an optimization model for
speed control, while Knorr et al. (2011) estimated the benefits which can be achieved from this speed control (and from adjusting flight descent profiles) to be large.

Finally, relating to the airport surface management subproblem (d), Marín (2006); Rathinam, Montoya, and Jung (2008); and Malik, Gupta, and Jung (2010) proposed approaches to the optimization of aircraft taxi routes, whereas Gotteland, Deau, and Durand (2009), and more recently Clare and Richards (2011), 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.

1.2. Contributions of the Paper

In this 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 that solves subproblems (a)–(d) 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 naive 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 suboptimal, or indeed infeasible.

The most notable aspects of our modeling approach are the following:

(i) Our definition of decision variables for runway sequencing leads to a greatly reduced state-space. This is achieved by capitalizing on the sole dependence of the minimum separation between two flights at a single runway 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 data sets for two international airports, Boston Logan International (BOS) and Dallas/Fort Worth International (DFW), which weighs in significant evidence to support first the claim of computational tractability and second the claim that our optimization can provide significant benefits for air traffic systems.

We note, however, several points related to the potential for direct implementation of the model presented in this paper. First, we assume that all data inputs are 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 might be used. Indeed, there are various possible sources of uncertainty that should be considered before attempting to implement the results directly, such as in runway availability (due, for example, to wind variation) or in the times at which departures are available for pushback from the gate. Second, we do not consider the aspect of fairness between the various agents involved—it is a possibility, for example, that different airlines may be treated differently by a “system-optimal” solution. Third, the model falls under a central decision maker paradigm, such as was considered by Hall (1999). In his case, an air navigation service provider (such as the FAA) would act as a central decision-maker and dictate arrival and departure rates at each airport within an air traffic network, being endowed with full information from each airport. In our case, the airport controller is the central decision maker and the airlines are the other key stakeholders. Although such a paradigm is conducive to gaining solutions that are theoretically optimal for the system as a whole, it does not respect the current tenet of collaborative air traffic management (ATM) under which decisions should lie with individual stakeholders as much as possible. For example, in the current ATM environment an airline can largely choose when to push back a departure based on its own preferences—the precise pushback time is not dictated by the airport controllers. Finally, our model incorporates all decisions into one optimization, which may pose implementation challenges because of the difference in the time scales at which the decisions are currently made. For example, runway configuration changes are made much less frequently than runway sequencing decisions.

The above considerations represent barriers to directly implementing the full set of decisions proposed by our model. However, it could certainly be possible to implement a subset of its decisions (such as configuration sequences, for example) under the current ATM framework. Further, the model can be of significant benefit for analysis purposes, demonstrating how a highly efficient airport might run, enabling lessons to be drawn that can influence future operations. Finally, we strongly believe that the methodology we present will have increasing practical value into the future as ATM practices are improved to increase collaboration between stakeholders and work toward solutions that are closer to optimal in a systemwide sense.
1.3. Structure of the Paper
This paper is structured as follows:
- In §2, we present phase one of our two-phase approach to optimizing the entirety of key ATFM 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 §3, we present phase two of the methodology, which uses the solution from phase one to tractably solve the problem under an assumption that is much milder than that of phase one and that is indeed a very realistic one.
- In §4, we present extensive computational experience based on real-world historic data sets at DFW and BOS to demonstrate both that our methodology is computationally tractable and that it can present significant benefits in practice.

2. The Airport Operations Optimization Problem
In this section we present a novel binary optimization model that represents phase one of our two-phase approach to solve the entirety of key ATFM decisions to be made at an airport and within its near-terminal airspace, which we shall call 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; and
- (iii) a route from gate to assigned runway, and then to departure fix, with timing, as well as 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; and
- (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 nonbinding. 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 phase two optimization problem that is relatively easy to solve. The solution to this second phase of optimization is guaranteed to be feasible for the AOOP, provided flight takeoff/landing times do not have strict upper time windows, which is usually 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), whereas 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 complementary 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 §4 will show there to be almost no loss of optimality in the real-world instances to which we apply our methodology.

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 $i = (w, o)$ 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 constituting the time horizon considered;
- $\mathcal{C} = \{\text{set of flight types} \}$, each of which is a pair $i = (w, o)$ corresponding to a weight class category $w$ and a flight orientation (arrival/departure) $o$;
- $\mathcal{C}_A, \mathcal{C}_D = \{\text{set of flight types whose orientation is arrival and departure, respectively}\}$;
- $\mathcal{F} = \mathcal{F}_A \cup \mathcal{F}_D = \bigcup_{i \in \mathcal{C}} \mathcal{F}_i = \{\text{set of flights}\}$;
- $\mathcal{c}_f(\in \mathcal{C}) = \{\text{type of flight}\}$;
- $\mathcal{R} = \{\text{set of runways, each of which includes a single, fixed direction of operation}\}$;
\( R_f, R_i = \) the set of runways that is feasible for flight \( f \), or for some flight of type \( i \), 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 \( \{(r_i^1, r_j^1), \ldots, (r_i^N, r_j^N)\} \) 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 \( k = \{(r_1, m_1), \ldots, (r_N, m_N)\} \), where \( m_i \) 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;

\( \mathcal{R}_k = \) the set of runways used by configuration \( k \);

\( \mathcal{F}_{rk} = \) the set of flight types that can use runway \( r \) under configuration \( k \);

\( \mathcal{U}_i(\subseteq \mathcal{R}) = \) those runways that cannot be used at time \( t \) because of, for example, strong crosswinds to tailwinds;

\( \mathcal{T}_f^r = \{T_f^r', T_f^r' + 1, \ldots, T_f^r\} \) = the set of feasible times for flight \( f \) to arrive at runway \( r \), considering the flight’s earliest possible pushback time (or arrival fix time) and location and the shortest paths to and from \( r \), when unimpeded by traffic. We refer to the desired upper bound on the takeoff/landing time, \( T_f^r' \), as the flight’s “deadline,” although this is assumed not to be strict;

\( T_f^r' \) = the release time of flight \( f \) from its origin (gate or arrival fix) into the system;

\( l_t^i \) = the number of time intervals constituting the runway occupancy time of flights of type \( i \) at runway \( r \);

\( s_{ij}^t = \) 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 de Neufville and Odoni (2003) for more details, but we note that this is always at least equal to \( l_t^i \), the runway occupancy time of the first aircraft;

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

\( \beta_i^{g} = \) the cost per time interval for a flight of type \( i \) at the gate before pushback (if a departure) in the air before reaching arrival fix (if an arrival);

\( \beta_i^{f} = \) the cost per time interval for a flight of type \( i \) traveling between its origin (gate/arrival fix) and its assigned runway;

\( \beta_i^{R} = \) the cost per time interval for a flight of type \( i \) traveling between its assigned runway and its destination (departure fix/gate);

\( d_{fr}^{(2)} = \) the distance of a shortest path for flight \( f \) from its origin (gate/arrival fix) to runway \( r \), in case (1), or from runway \( r \) to its destination (departure fix/gate), in case (2);

\( K = \) a constant that penalizes each configuration changeover.

### 2.2. Decision Variables

We define the following binary decision variables for our model:

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

\[ \varphi_{fr} = \begin{cases} 1, & \text{if flight } f \text{ is assigned to runway } r, \\ 0, & \text{otherwise}; \end{cases} \]

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

\[ \chi_{fr} = \begin{cases} 1, & \text{if a change of configuration occurs at time } t, \\ 0, & \text{otherwise}. \end{cases} \]

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

### 2.3. Objective Function

Our objective is to minimize function (1), which represents a weighted summation of flight travel costs

\[
\Psi \doteq \sum_{t \in \mathcal{T}} \sum_{r \in \mathcal{R}_{rk}} \sum_{i \in \mathcal{F}_{rk}} t \psi_{fr}^i - \sum_{f \in \mathcal{F}} \sum_{r \in \mathcal{R}_{rk}} T_f^r \varphi_{fr}^i + \sum_{f \in \mathcal{F}} \beta_i^{g} d_{fr}^g + \beta_i^{f} d_{fr}^f + K \sum_{t \in \mathcal{T}} \chi_{fr} \quad (1)
\]

This can be summarized as a summation over all flights of (i) the cost of time spent at the flight’s origin (at gate with engines off/before reaching arrival...
fix) before entering the system; (ii) the cost of travel along the shortest path between origin, assigned runway, and destination; and (iii) a configuration change penalty, which prevents too many from occurring because this is undesirable in practice. Note 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. Similarly, if arrivals reach their assigned runway any later than the earliest possible time, then they are speed controlled before reaching the fix to make up the slack duration. Because we are modeling by flight type in the phase one optimization problem, we are not able to capture flight-specific objective function coefficients here, unlike in phase two.

2.4. The Binary Optimization Problem

A binary optimization problem for the AOOP under nonbinding network capacity is then the following:

\[
\text{P1: } \min \Psi
\]

s.t. \[\sum_{k \in \mathbb{K}} \omega_{k_t} = 1, \ \forall t \in \mathcal{T}, \quad (2a)\]

\[\phi_{r_t}^i = 0, \ \forall i \in \mathcal{E}, r \in \mathcal{R}_t, t \in \mathcal{T}, \quad (2b)\]

\[\psi_{r_t, i-h}^j + \psi_{r_t}^j \leq 1, \ \forall i, j \in \mathcal{E}, r \in \mathcal{R}_i \cap \mathcal{R}_j, h \in \{1, \ldots, \min\{s_{ij}^r - 1, t-1\}\}, t \in \mathcal{T} \setminus \{1\}, \quad (2c)\]

\[\psi_{r_t, i-h}^j + \psi_{r_t}^j \leq 1, \ \forall i, j \in \mathcal{E}, (r, r') \in (\mathcal{R}_i \times \mathcal{R}_j) \cap \mathcal{E}, \quad (2d)\]

\[\sum_{i \in \mathcal{E}, r \in \mathcal{R}_i} \psi_{r_t}^i \leq 1, \ \forall r \in \mathcal{R}, t \in \mathcal{T}, \quad (2e)\]

\[\psi_{r_t}^i + \omega_{k_t} \leq 1, \ \forall t \in \mathcal{T}, i \in \mathcal{E}, r \in \mathcal{R}_k, \quad (2f)\]

\[h \in [0, \ldots, \min\{s_{ij}^r, s_{ij}^{r'} - 1, t-1\}], t \in \mathcal{T}, \quad (2g)\]

\[\sum_{r \in \mathcal{R}_i} \varphi_{r}^f \leq 1, \ \forall f \in \mathcal{F}, \quad (2h)\]

\[\sum_{f \in \mathcal{F}, r \in \mathcal{R}_f, t \in \mathcal{T}} \varphi_{r}^f \leq \sum_{f \in \mathcal{F}, r \in \mathcal{R}_f, t \in \mathcal{T}} \varphi_{r}^f, \quad (2i)\]

\[\chi_t - \omega_{k_t} + \omega_{k_t, t-1} \geq 0, \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \setminus \{1\}, \quad (2j)\]

\[\phi_{r_t}^i \in [0,1], \ \forall r \in \mathcal{R}_i, t \in \mathcal{T}, \quad (2m)\]

\[\chi_t \in [0,1], \ \forall t \in \mathcal{T}. \quad (2n)\]

Constraints (2a) require exactly one configuration to be used at any time, whereas Constraints (2b) prevent flights from occupying runways that 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 \(\omega\) variables) that 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 “subconfiguration” of a configuration does not require additional configuration variables). In addition, it enables us to add an extra class of valid inequalities (Inequalities (3), detailed in Proposition 1) to strengthen the model.

Constraints (2c) can generally be referred to as the separation constraints, which state that if a flight of type \(i\) takes off/lands, then we must wait at least \(s_{ij}^r\) time periods before a flight of type \(j\) takes off/lands on any given runway. An important point to note here is that these constraints correctly model 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 (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. In this paper we 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 five nautical miles, as in de Neufville and Odoni (2003). Indeed, any miles-in-trail or time-based separation requirement in the near-terminal airspace between fix and runway could be included in the same way—as a modification to the runway separation requirements in phase one and then enforced at the fix in phase two. When flights use different runways and/or different fixes, the relevant airspace separation requirements will be enforced in our second-stage problem. Because of the nature of runway configurations, where a general flow
in one direction is typically preserved and arrival and departure fixes are often procedurally segregated, these second-stage constraints would not typically be expected to have a significant impact on the overall optimality of our two-phase approach. We also note here that one other alternative is to extend the definition of flight type in the phase one problem to include a fix.

We now remind the reader that the definition of $\Psi$ is such that $\psi_{ij}^{rt} = 1$ 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 $\Psi$. Then Constraints (2e) state that only one flight may arrive at each runway at any given time. This set of constraints, along with Constraints (2c), enforce the capacity of each runway to be one at all times (recall $s_{ij}^r \geq l_r^i$). Constraints (2f) disallow the use of runway $r$ for flights of type $i$ if such use is not allowed under configuration $k$. Constraints (2g) state that if we process a flight of type $i$ at a given runway $r$, then that runway must remain open for at least $l_r^i$ time periods, corresponding to the runway occupancy time of flights of type $i$.

Constraints (2h) state that every flight must be assigned to some runway. Then Constraints (2i) require each flight $f$ to be processed at one of its feasible runways $r$ after its earliest possible touchdown/takeoff time $T_{ij}^r$. The left-hand side is equal to the number of flights assigned to runway $r$ that 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 $r$ that could feasibly have arrived at $r$ by time $t$ (recall this is based on shortest paths). Thus, 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 $t \in T$. These are the only constraints that link the $\Psi$ variables with the $\varphi$ variables.

Finally, Constraints (2j) enforce $\chi_i = 1$ if a change of configuration occurs at time $t$, which happens if and only if $\exists k$: $\omega_{kt} = 1$ and $\omega_{kt-1} = 0$. Note that this is equivalent to setting $\chi_i = 1$ since $\chi$ is penalized in the objective function and this is the only constraint on $\chi$.

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. Then P1 gives an optimal solution to the AOOP, including the optimal configuration schedule (through $\omega$), the optimal runway assignments ($\varphi$), the optimal sequencing of flights ($\Psi$), 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; and
  (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 $\varphi_{ij}^{rt}$ 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 grew, especially because of 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, 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 $\psi_{ij}^{rt} = 1$ 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.

• Because the variables $\Psi$ 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 $\varphi$ we also have an assignment of flights to runways, and it is through the time window constraints (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 Anagnostakis and Clarke (2003) and Solveling et al. (2011). Inspection of these constraints reveals that there is always at least one sequence of flights corresponding to a solution of P1 when the latest takeoff/touchdown times of flights are not strict 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.

2.5. Valid Inequalities

We can strengthen the formulation P1 by adding certain valid inequalities. Let $\bar{s}_j = \min_{i \in \xi} \{s_{ij}^r\}$ be the minimum possible separation time required between two flights at runway $r$ when the second flight is of type $j$. Also let $\mathcal{X}$ be the set of feasible solutions to P1 that
have the additional property that \( \forall t \in \mathcal{T} \setminus \{1\}, \exists k: \chi_t = \omega_{kt} - \omega_{kt-1} \geq 0 \). Because \( \chi_t \) is penalized in the objective function, any optimal solution has this property. Thus we have the following proposition:

**Proposition 1.** The following inequalities are valid for the polyhedron \( \text{conv}(\mathcal{X}) \). Furthermore, they are not valid for the linear relaxation of P1.

\[
\chi_t - \omega_{kt-1} \geq 0, \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \setminus \{1\},
\]

\[
\chi_t + \omega_{kt-1} + \omega_{kt} \leq 2, \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \setminus \{1\},
\]

\[
\sum_{i \in r \in \mathcal{R}_i} \psi_{t, t+h} \leq 1, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}.
\]

Proof. See the online appendix (available as supplemental material at [http://dx.doi.org/10.1287/trsc.2015.0590](http://dx.doi.org/10.1287/trsc.2015.0590)).

3. **Phase Two—The Routing Subproblem**

In this section we detail phase two 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 schedule that is very close to that obtained in phase one, if not the same. In particular, we fix the solution from phase one to subproblems (a) and (b) outlined in §2, and here in phase two 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 that 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 takeoff/landing times do not have strict deadlines while also ensuring the solution retains the key properties of the phase one solution. In the case of infeasibility with the flight deadlines, we would require these deadlines 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 phase two. In §4, we shall support this statement.

3.1. **Data**

To model the routing subproblem, the airport network is represented by a directed graph with nodes belonging to the set \( \mathcal{F} \), where each node represents a section of taxiway, a runway, an airspace route, a gate, or a fix. See the online appendix for simplified diagrams of such networks for DFW. A list of relevant sets and parameters, building on those of §2, is given below:

\( \mathcal{F} \) = the set of nodes in the airport network;

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

\( \mathcal{R}_i \) = the set of nodes that are successors of node \( i \) for flight \( f \);

\( \mathcal{P}_i \) = the set of nodes that are predecessors of node \( i \) for flight \( f \);

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

\( \mathcal{T}_f^i = \{ \bar{T}_f^i, \bar{T}_f^i + 1, \ldots, \bar{T}_f^i \} \) = the set of feasible times for flight \( f \) to arrive at node \( i \), considering the flight’s earliest possible pushback/arrival-fix time and location and the shortest path to \( i \), when unimpeded by traffic;

\( o_f \) = initial node of flight \( f \);

\( l_f^i \) = the minimum amount of time flight \( f \) must spend at node \( i \);

\( u_{ii} \) = the capacity of node \( i \), in flights, at time \( t \).

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 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 among flights of the same type which are assigned to the same runway, and only ones which respect the time window constraints. Although one can imagine many possible schemes for determining this ordering, this is not a focus of this paper and we shall now assume we have fixed such an ordering.

\( r_f \) = the assigned runway node at which flight \( f \) should be processed;

\( \mathcal{E} = \{(f_1, g_1), \ldots, (f_n, g_n)\} \) = the set of pairs of flights \((f, g)\) such that the following hold:

(i) flight \( f \) is scheduled to be processed on runway \( r \) in configuration \( A \);

(ii) flight \( g \) is scheduled to be processed on runway \( q \) in configuration \( B \);

(iii) configuration \( A \) is scheduled to be used 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;
3.3. Decision Variables

We have the following decision variables:

\[ z_{ji}^f = \begin{cases} 1, & \text{if flight } f \text{ reaches node } i \text{ by time } t, \\ 0, & \text{otherwise}; \end{cases} \]

\[ x_{ji}^f = \begin{cases} 1, & \text{if flight } f \text{ is at node } i \text{ at time } t, \\ 0, & \text{otherwise}. \end{cases} \]

Note that the \( z \) variables are defined as “by” variables in the spirit of Bertsimas and Stock-Patterson (1998), which will lead to nice properties in the model formulation.

3.4. Objective Function

The objective function that we minimize captures 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—and is described by function (6)

\[
\Phi \triangleq \sum_{j \in \mathcal{J}} \left[ \beta_{e_j}^0 \left( \gamma_j^f - \sum_{i \in \mathcal{I}_j^f} z_{ji}^f - l_{e_j}^f \right) \right. \\
+ \beta_{e_j}^1 \left( l_{e_j}^f + \sum_{i \in \mathcal{I}_j^f} \gamma_i^f - \sum_{j \in \mathcal{N}_i} \gamma_j^f \right) \\
+ \left. \beta_{e_j}^2 \left( \sum_{j \in \mathcal{N}_i} \gamma_j^f - \sum_{j \in \mathcal{N}_i} \gamma_j^f \right) \right],
\]

where \( \gamma_j^f = \sum_{t=1}^{T} z_{ij}^f \) 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

\[ \gamma_i^f - \gamma_j^f = \sum_{t=1}^{T} z_{ij}^f - \sum_{t=1}^{T} z_{ji}^f \]

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 (6) 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 \( l_{e_j}^f \) is the time it takes until a departure \( f \) begins taxiing after removing its brakes (and 0 for an arrival) and is assumed to be constant in this paper.

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 that were found in P1. The model is based on the models of Bertsimas, Lulli, and Odoni (2008) and Bertsimas and Stock-Patterson (1998), which were presented to solve the network ATFM problem with and without rerouting, respectively.

\[ \text{P2: } \min \Phi \]

\[ \text{s.t. } x_{ji}^f - \left( z_{ji}^f - \sum_{i \in \mathcal{I}_j^f} z_{ji}^f \right) \geq 0, \quad \forall f \in \mathcal{F}, j \in \mathcal{N}_f, t \in \mathcal{T}_f^j, \tag{7a} \]

\[ \sum_{f \in \mathcal{F}, j \in \mathcal{N}_f} x_{ji}^f \leq u_{ji}, \quad \forall j \in \mathcal{N} \setminus \mathcal{R}, t \in \mathcal{T}, \tag{7b} \]
\[ z^f_{i,t} - \sum_{j \in \mathcal{G}_i} z^f_{i,t-1,j} \leq 0, \quad \forall f \in \mathcal{T}, \quad t \geq t^f, \]  
(7a)

\[ \sum_{j \in \mathcal{G}_i} z^f_{i,t} = 1, \quad \forall f \in \mathcal{T}, \]  
(7e)

\[ z^f_{i,t-1} - z^f_{i,t} \leq 0, \quad \forall f \in \mathcal{T}, \quad t \geq t^f, \]  
(7b)

\[ z^f_{i,t} - \sum_{j \in \mathcal{G}_i} z^f_{j,t} \leq 0, \quad \forall f \in \mathcal{T}, \quad i \in \mathcal{F} \setminus \{o_f\}, \]  
(7c)

\[ \sum_{j \in \mathcal{G}_i} z^f_{j,t} \leq 1, \quad \forall f \in \mathcal{T}, \quad i \in \mathcal{F}, \]  
(7f)

\[ z^f_{i,t-1} - z^f_{i,t} \leq 0, \quad \forall f \in \mathcal{T}, \quad j \in \mathcal{F} \setminus \{o_f\}, \]  
(7g)

Constraints (7a) link the \( x \) variables with the \( z \) variables, with \( x^f_{i,t} \) 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 (7b) 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). Observe that these constraints ensure that gates are capacitated.

Constraints (7c) state that flight \( f \) cannot reach a node \( j \) by time \( t \) unless it has reached one of its predecessors \( i \) by time \( t - t^f \). Constraints (7d) require that a flight \( f \) must eventually reach some follower of any node it reaches unless that node is its destination, in which case Constraints (7e) state that the flight must reach one of its feasible destinations. Constraints (7f) 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 (7g) enforce monotonicity on the \( z \) variables, owing to their definition. Constraints (7h) initialize each flight at its origin.

Constraints (7i) force a flight to use its assigned runway from P1. Constraints (7j) 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, whereas Constraints (7k) enforce these same ordering and separation requirements for all pairs of runways scheduled on intersecting/closely spaced parallel runways. Constraints (7l) ensure that flights are adequately separated at their arrival/departure fix. Note that the separation is already incorporated in phase one for those flights that use a common runway but may also be included here. Note also that since arrivals and departures use separate fixes in general, the number of such constraints will be small. All three sets of constraints (7j)–(7l) are of a much nicer form than usual separation constraints, for two reasons. First, there is 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 \( \mathcal{H}_r \), \( \mathcal{H}_r(r,r) \), and \( \mathcal{W} \). Second, the form of the constraints states that one set of the “by” variables \( z \) must dominate another set by a specified amount. Indeed, in Bertsimas and Stock-Patterson (1998) 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 (7m) ensure that the configuration requirements are respected by ensuring that we process all pairs of flights in the set \( \mathcal{E} \) in the specified order. Note that we have defined the set \( \mathcal{E} \) to be as small as possible while still preventing the operation of illegal configurations, expanding the feasible space of P2. Observe that when a configuration change occurs, there will typically be some lost capacity in the short term since for flights to be feasibly routed to their assigned runways immediately after the configuration change, they may need to wait for some taxiways/airways being used from the previous configuration to have cleared.
Constraints (7n) state that a flight may not be processed at a given runway when that runway is not available (for example, because of weather conditions). Finally, Constraints (7o) 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 (7a) 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, because of the nonexistence of the variable z_{jt}^f.

3.6. Valid Inequalities
We add the following valid inequalities to strengthen the model P2:

\[ \sum_{i \in A_f'} \bar{z}_{ji}^f \leq 1, \quad \forall f \in \mathcal{F}, \forall t \in \mathcal{T} \setminus \{0\}, z_{ji}^f \geq 0, \forall f \in \mathcal{F}, j \in \mathcal{F}_f, \quad \forall t \in \mathcal{T}_f \setminus \{0\}, \tag{8} \]

\[ \sum_{i \in A_f'} \bar{z}_{ji}^f \geq 0, \quad \forall f \in \mathcal{F}, j \in \mathcal{F}_f, \quad \forall t \in \mathcal{T}_f \setminus \{0\}, \tag{9} \]

\[ \sum_{i \in A_f'} \bar{z}_{ij}^f \leq 1, \quad \forall f \in \mathcal{F}, \forall \text{antichains } A_f \subset \mathcal{F}_f. \tag{10} \]

Inequalities (8) and (9) 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 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 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 (10) state that if there is a set A ⊂ F_f of nodes for which, based on the structure of the network, we must have \( \sum_{i \in A} \bar{1}(\text{flight } f \text{ reaches } i) \leq 1 \), then only one of the corresponding variables \( z_{ij}^f \), \( j \in A \), can be nonzero. All Inequalities (8)–(10) were introduced in Bertsimas, Lulli, and Odoni (2008), and we direct the reader to that paper for further details.

4. Computational Experiments
In this section we present extensive computational experiments that seek 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?
- Could the use of the methodology result in significant benefits in practice?

To answer these questions, we focus on two international airports: BOS and DFW. We utilize data from historic days of operation at these airports, November 2, 2009 at DFW and September 28, 2010 at BOS. In particular, we make use of the following data sources:

- Hourly METAR (see Plymouth State University 2010) weather forecasts, from which runway availability is determined;
- Airport Surface Detection Equipment, Model X (ASDE-X) flight track data, indicating individual flight positions and timing, velocity, acceleration, and heading, among other fields; and
- 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 Gurobi 5.0 on a computer with an Intel Core i7-860 Processor (8 MB, 2.8 GHz) and 16 GB DDR3 RAM, running Ubuntu Linux 10.04. The solver time limit was set to 1,200 seconds for each of P1 and P2.

For each historic time period considered, we input the corresponding weather, flight schedules, etc., as data to our model. In all of the tables that follow, our time horizon incorporates one hour worth of arrivals and departures, and we use a discretization of 20 seconds per time interval. The values that most parameters take should be self-explanatory, but we note how we set the following parameters:

\[ \mathcal{T}_{0f} := \begin{cases} 
\text{the actual pushback time of flight } f, & \text{if } f \text{ is a departure,} \\
\text{the actual touchdown time of flight } f \text{ less a constant,} & \text{if } f \text{ is an arrival;}
\end{cases} \]

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

4.1. Model Validation and Computational Tractability
In this section we set \((\beta_D^0, \beta_D^1, \beta_A^0) = (0.5, 1.1, 1.3)\) and \((\beta_A^1, \beta_A^2) = (1.2, 1.2, 1.0)\) to resemble a natural objective, which prioritizes reducing times in the air more than those on the ground and which also prioritizes departure throughput more than arrival throughput. Further, there is an incentive to gate-hold departures, but no incentive to speed control arrivals before they reach their arrival fix.

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 optimal values from phase one and phase 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 suboptimality of P2. This indicates that our two-phase approach results in solutions that 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. One can observe the consistency of the P1 computation times, but also that in some instances, P2 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.

### 4.2. Benefits Assessment

Above we have demonstrated that our approach leads to solutions that 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 mean and standard deviation of the times taken for flights to traverse part of the system across several one-hour runs on the same day at DFW and BOS. For arrivals we record the time from touchdown until arrival at the gate, and for departures we record the time from pushback until takeoff. Ideally, we would present the overall system traversal times, from fix to gate or from gate to fix, but because of a 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. Figures 2–5 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% to 14% at DFW and 7% to 25% at BOS. For arrivals, however, surface traversal times are in general worse in the optimized solution because of 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 (although in Table 5, we demonstrate that delays may be transferred by altering the objective function weights). 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 nonnegligible 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>
<thead>
<tr>
<th>Number of flights</th>
<th>Objective values</th>
<th>% difference</th>
<th>% optimality gap</th>
<th>Computation times (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>155</td>
<td>12,806.2</td>
<td>13,073.6</td>
<td>2.1</td>
<td>0.0</td>
</tr>
<tr>
<td>175</td>
<td>14,365.3</td>
<td>14,530.4</td>
<td>1.1</td>
<td>0.0</td>
</tr>
<tr>
<td>153</td>
<td>12,513.7</td>
<td>12,586.7</td>
<td>0.6</td>
<td>0.0</td>
</tr>
<tr>
<td>155</td>
<td>12,879.2</td>
<td>12,972.0</td>
<td>0.7</td>
<td>0.0</td>
</tr>
<tr>
<td>168</td>
<td>13,988.0</td>
<td>14,064.8</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>171</td>
<td>14,584.6</td>
<td>14,683.0</td>
<td>0.7</td>
<td>0.0</td>
</tr>
<tr>
<td>159</td>
<td>13,018.4</td>
<td>13,140.9</td>
<td>0.9</td>
<td>0.0</td>
</tr>
<tr>
<td>153</td>
<td>12,483.2</td>
<td>12,556.2</td>
<td>0.6</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Note.** The P1 objective value has had the configuration change penalty removed.

<table>
<thead>
<tr>
<th>Number of flights</th>
<th>Objective values</th>
<th>% difference</th>
<th>% optimality gap</th>
<th>Computation times (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>4,687.6</td>
<td>4,696.3</td>
<td>0.2</td>
<td>0.0</td>
</tr>
<tr>
<td>91</td>
<td>4,619.8</td>
<td>4,639.3</td>
<td>0.4</td>
<td>0.0</td>
</tr>
<tr>
<td>80</td>
<td>5,114.7</td>
<td>5,117.7</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>63</td>
<td>3,111.2</td>
<td>3,128.7</td>
<td>0.6</td>
<td>0.0</td>
</tr>
<tr>
<td>63</td>
<td>3,168.2</td>
<td>3,179.2</td>
<td>0.3</td>
<td>0.0</td>
</tr>
<tr>
<td>71</td>
<td>3,877.1</td>
<td>3,928.3</td>
<td>1.3</td>
<td>0.0</td>
</tr>
<tr>
<td>59</td>
<td>2,681.4</td>
<td>2,695.0</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>63</td>
<td>3,512.9</td>
<td>3,524.4</td>
<td>0.3</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Note.** The P1 objective value has had the configuration change penalty removed.
Table 3  Comparison of Optimized and Historic Surface Times at DFW

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>155</td>
<td>1.8±2.7</td>
<td>9.5±1.8</td>
<td>10.8±2.6</td>
<td>10.2±2.3</td>
<td>13.0±3.9</td>
<td>8.9±3.9</td>
<td>10.7±4.4</td>
<td></td>
</tr>
<tr>
<td>175</td>
<td>2.2±3.2</td>
<td>9.2±1.9</td>
<td>10.7±2.4</td>
<td>9.9±2.2</td>
<td>12.5±3.2</td>
<td>9.3±4.4</td>
<td>11.2±4.2</td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>1.0±1.8</td>
<td>9.5±2.0</td>
<td>10.6±2.3</td>
<td>10.0±2.2</td>
<td>13.5±3.0</td>
<td>9.2±4.9</td>
<td>11.6±4.5</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>1.0±1.2</td>
<td>9.6±1.9</td>
<td>11.1±2.4</td>
<td>10.4±2.3</td>
<td>13.5±3.9</td>
<td>9.0±3.7</td>
<td>11.0±4.4</td>
<td></td>
</tr>
<tr>
<td>168</td>
<td>0.9±1.4</td>
<td>9.3±1.9</td>
<td>11.8±2.8</td>
<td>10.5±2.7</td>
<td>13.0±2.9</td>
<td>10.1±5.7</td>
<td>11.6±4.7</td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>2.1±3.0</td>
<td>10.0±2.1</td>
<td>11.4±2.5</td>
<td>10.7±2.4</td>
<td>13.6±3.2</td>
<td>8.9±3.9</td>
<td>11.3±4.3</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>1.3±1.6</td>
<td>9.1±1.8</td>
<td>10.8±2.4</td>
<td>10.0±2.3</td>
<td>13.9±3.1</td>
<td>9.1±3.9</td>
<td>11.4±4.3</td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>0.7±1.0</td>
<td>9.0±1.8</td>
<td>11.1±2.1</td>
<td>10.1±2.2</td>
<td>13.4±3.8</td>
<td>9.6±3.6</td>
<td>11.2±4.1</td>
<td></td>
</tr>
</tbody>
</table>

Notes. Statistics given are the relevant mean and standard deviation, in minutes per flight for gate-holding, from pushback to wheels-off for departures, and from wheels-on to gate-in for arrivals.

Table 4  Comparison of Optimized and Historic Surface Times at BOS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>1.6±2.0</td>
<td>13.7±0.8</td>
<td>4.4±1.2</td>
<td>9.3±4.8</td>
<td>18.7±6.0</td>
<td>5.2±1.1</td>
<td>12.4±8.1</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>0.8±1.0</td>
<td>14.0±0.8</td>
<td>4.5±1.2</td>
<td>9.6±4.8</td>
<td>16.7±4.8</td>
<td>6.1±1.7</td>
<td>11.6±6.5</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>2.2±3.9</td>
<td>16.4±0.7</td>
<td>5.0±1.1</td>
<td>12.2±5.6</td>
<td>17.0±5.3</td>
<td>5.9±2.0</td>
<td>13.1±7.2</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>0.5±0.7</td>
<td>16.6±0.6</td>
<td>4.9±1.1</td>
<td>10.3±5.9</td>
<td>18.2±8.4</td>
<td>6.5±1.7</td>
<td>12.2±8.4</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>0.5±0.7</td>
<td>16.0±0.8</td>
<td>5.0±1.1</td>
<td>10.5±5.6</td>
<td>18.5±8.2</td>
<td>4.4±1.1</td>
<td>10.9±9.0</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>1.3±1.5</td>
<td>11.6±1.0</td>
<td>8.2±0.8</td>
<td>10.1±1.9</td>
<td>16.6±4.4</td>
<td>5.1±0.9</td>
<td>11.4±6.6</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>2.6±5.2</td>
<td>13.8±0.7</td>
<td>4.6±1.3</td>
<td>8.5±4.7</td>
<td>14.7±4.9</td>
<td>5.8±1.3</td>
<td>9.9±5.7</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>0.9±1.1</td>
<td>16.2±0.9</td>
<td>4.9±1.1</td>
<td>10.5±5.7</td>
<td>17.1±6.9</td>
<td>6.6±1.8</td>
<td>11.3±7.1</td>
<td></td>
</tr>
</tbody>
</table>

Notes. Statistics given are the relevant mean and standard deviation, in minutes per flight for gate-holding, from pushback to wheels-off for departures, and from wheels-on to gate-in for arrivals.

Figures 6 and 7 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 that 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 on the x-axis of the arrival congestion between the historic and optimized data reflects that exact fix-times were not available as inputs to our optimization, 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 by 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 because of two phe-
nomena present in the optimal policies—first, almost any delays to departures are incurred at the gate with engines off; second, aircraft do not queue up at runways but rather leave the gate at just the right time to reach their assigned runway for takeoff. Both phenomena, although desirable in an ideal world, may not always be practical, however. In the former case, since our model works under a central decision maker paradigm and does not consider individual airline preferences nor equity constraints, the exact nature of the solution—which flights are held back and which push back—may not necessarily satisfy airlines, and future work would be needed to incorporate these requirements. In the latter case, minimal queueing at runways is optimal because of our assumption that the input data are deterministically known. In future work, the incorporation of robustness to delays in aircraft availability would be expected to add a small buffer to these optimized taxi times.

4.3. Further Analysis

Table 5 demonstrates the effect of changing the objective function coefficients. On the left-hand side of the table, we have the weights from §4.2, favoring reducing times of departures in the air, then arrivals in the air, then departures on the ground, then arrivals on the ground. Further, there is an incentive to gate-hold departures, but not to speed control arrivals before reaching their fix. On the right-hand side of the table, we favor reducing times of arrivals in the air, then departures on the ground, then arrivals on the ground. We also add an incentive to delay arrivals before reaching their fix. We can see that this has a significant effect—transferring delays from arrivals to departures.

Table 6 shows the effect of different numbers of configuration changes on optimized surface times at DFW. These results were obtained by forcing the number of configuration changes in the optimization. As expected, an overall decrease in surface times.
Table 5  Comparison of Optimized Surface Times Favoring Departures (Left) vs. Favoring Arrivals (Right) at DFW

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>155</td>
<td>1.8±2.7</td>
<td>9.5±1.8</td>
<td>10.8±2.6</td>
<td>10.2±2.3</td>
<td>1.4±2.6</td>
<td>13.0±2.3</td>
<td>7.5±3.0</td>
<td>9.9±3.9</td>
</tr>
<tr>
<td>175</td>
<td>2.2±3.2</td>
<td>9.2±1.9</td>
<td>10.7±2.4</td>
<td>9.9±2.2</td>
<td>1.6±2.5</td>
<td>9.2±1.9</td>
<td>10.7±2.4</td>
<td>9.9±2.3</td>
</tr>
<tr>
<td>153</td>
<td>1.0±1.8</td>
<td>9.5±2.0</td>
<td>10.6±2.3</td>
<td>10.0±2.2</td>
<td>0.7±1.3</td>
<td>9.5±2.0</td>
<td>10.6±2.3</td>
<td>10.0±2.2</td>
</tr>
<tr>
<td>155</td>
<td>1.0±1.2</td>
<td>9.6±1.9</td>
<td>11.1±2.4</td>
<td>10.4±2.3</td>
<td>0.9±1.6</td>
<td>12.9±2.3</td>
<td>7.9±2.7</td>
<td>10.2±5.5</td>
</tr>
<tr>
<td>168</td>
<td>0.9±1.4</td>
<td>9.3±1.9</td>
<td>11.8±2.8</td>
<td>10.5±2.7</td>
<td>0.9±1.3</td>
<td>13.8±3.4</td>
<td>8.2±2.9</td>
<td>11.0±4.2</td>
</tr>
<tr>
<td>171</td>
<td>2.1±3.0</td>
<td>10.0±2.1</td>
<td>11.4±2.5</td>
<td>10.7±2.4</td>
<td>0.8±1.2</td>
<td>13.5±1.8</td>
<td>8.0±2.9</td>
<td>10.7±3.7</td>
</tr>
<tr>
<td>159</td>
<td>1.3±1.6</td>
<td>9.1±1.8</td>
<td>10.8±2.4</td>
<td>10.0±2.3</td>
<td>0.9±1.2</td>
<td>9.1±1.8</td>
<td>10.7±2.4</td>
<td>9.9±2.2</td>
</tr>
<tr>
<td>153</td>
<td>0.7±1.0</td>
<td>9.0±1.8</td>
<td>11.1±2.1</td>
<td>10.1±2.2</td>
<td>0.5±0.7</td>
<td>9.0±1.8</td>
<td>11.0±2.1</td>
<td>10.0±2.2</td>
</tr>
</tbody>
</table>

Notes. Statistics given are the relevant mean and standard deviation, in minutes per flight for gate-holding, from pushback to wheels-off for departures, and from wheels-on to gate-in for arrivals.

is seen as more configuration changes are allowed within the hour. This can also be seen in the density plots of optimized surface times in Figure 8.

Table 6  Comparison of Optimized Surface Times by Number of Configuration Changes at DFW

<table>
<thead>
<tr>
<th>Number of configuration changes</th>
<th>Number of flights</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>155</td>
<td>9.9±3.8</td>
</tr>
<tr>
<td>175</td>
<td>9.8±2.3</td>
</tr>
<tr>
<td>153</td>
<td>9.9±2.1</td>
</tr>
<tr>
<td>155</td>
<td>10.1±3.5</td>
</tr>
<tr>
<td>168</td>
<td>10.4±2.7</td>
</tr>
<tr>
<td>171</td>
<td>10.6±2.4</td>
</tr>
<tr>
<td>159</td>
<td>9.8±2.2</td>
</tr>
<tr>
<td>153</td>
<td>10.0±2.2</td>
</tr>
</tbody>
</table>

Notes. Here \((\beta^0_0, \beta^1_0, \beta^2_0) = (0.5, 1.5, 1.0)\) and \((\beta^3_0, \beta^4_0, \beta^5_0) = (0.5, 1.1, 1.4)\). Statistics given are the relevant mean and standard deviation, in minutes per flight for gate-holding, from pushback to wheels-off for departures, and from wheels-on to gate-in for arrivals.

Table 7 shows the effect of gate-holding. We can see that when gate-holding is disallowed, this results in a significant increase in surface travel times. This could be attributed to an increase in surface congestions.

Figure 7  Contribution of Scheduled Flights in the Optimization Time Window to Surface Congestion at BOS

Figure 8  (Color online) Effect of Configuration Changes on Optimized Surface Times at DFW
tion since flights are forced to leave the gate on time. To put the effect of gate-holding into perspective, the optimized surface times here are on average 10.1 minutes with gate-holding and 10.5 minutes without, compared to 11.3 minutes historically. This equates to a 6.5% improvement through optimization without gate-holding and an additional 4.1% improvement when gate-holding was also allowed in the optimization.

4.4. 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, with the complete optimization typically taking five to 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. It 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.

5. Conclusion

We have presented a novel, integrated approach to solving the entirety of key ATFM problems faced at an airport. Through computational experiments using historic data from BOS and DFW, we have shown the methodology to be both tractable (in the practical sense of having “reasonable” computation times—of the order of five to 10 minutes) and of significant potential benefit in terms of reduced delays, fuel use, and emissions. That said, as we indicated in §1, there are barriers to direct implementation of the model, especially under current ATM guidelines and practices. Nevertheless, the model certainly has the potential to influence ATM: (i) in the present day, a subset of its decisions could be useful for implementation, such as configuration selection; (ii) in the future, both as the ATM environment changes and as further work incorporates some of the key considerations we have not investigated in this paper, such as uncertainty and fairness, the model could pose increasing practical value; and (iii) the model could prove very useful as an analytical tool, from which valuable understanding could be drawn on how better to operate airports.

Acknowledgments

This work was supported by a grant from MIT Lincoln Laboratory. The authors would like to thank James Kuchar, Bill Moser, Tom Reynolds, Ngaire Underhill, and Mark Weber of MIT Lincoln Laboratory for their assistance with data, as well as Professor Amedeo Odoni, Ross Anderson, and Shubham Gupta of the MIT Operations Research Center, for helpful discussions. The authors would also like to extend their thanks to the reviewers for their insightful comments, from which this exposition has benefited.

References


