A Stochastic Integer Program with Dual Network Structure and its Application to the Ground Holding Problem

Michael O. Ball, Robert Hoffman*
Amedeo Odoni, Ryan Rifkin†

March 27, 2000

Abstract

In this paper, we analyze a generalization of a classic network flow model. The generalization involves the replacement of deterministic demand with stochastic demand. While this generalization destroys the original network structure, we show that the matrix underlying the stochastic model is dual network. Thus, the integer program associated with the stochastic model can be solved efficiently using network flow or linear programming techniques. We also develop an application of this model to the ground holding problem in air traffic management. The use of this model for the ground holding problem improves upon prior models by allowing for easy integration into the newly developed ground delay program procedures based on the Collaborative Decision Making paradigm.

*University of Maryland at College Park
†Massachusetts Institute of Technology

1 Introduction

In the paper, we analyze a stochastic variant of a classic network flow model. The (deterministic) network flow model is illustrated in Figure 1. The nodes across the top of the diagram are supply nodes and the nodes across the bottom are demand nodes. The model represents the progression of supply and demand for a commodity over discrete time intervals. The commodity can exist in one of two states. The top set of nodes are associated with the commodity in state 1 and the bottom set of nodes are associated with the commodity in state 2. Thus, node 1-1 represents the balance equation for the commodity during time period 1 in state 1, node 1-2 represents the balance equation for the commodity during time period 1 in state 2, etc. We denote by S_t and D_t for $t = 1, ..., T$, the supply and demand, respectively, during time period t. The model variables are defined by:

$$
\begin{aligned}
y_t &= \text{amount of commodity in state 1 held in inventory} \\
&\quad \text{from period } t \text{ to } t+1 \\
x_t &= \text{amount of commodity converted from state 1 to state 2} \\
&\quad \text{during time period } t \\
z_t &= \text{amount of commodity in state 2 held in inventory} \\
&\quad \text{from period } t \text{ to } t+1
\end{aligned}
$$

If we associate inventory holding costs a_t and r_t with the commodity in states 1 and 2 and a conversion cost c_t, we obtain the following network flow model:

$$\text{Min} \sum_{t=1}^{T} (a_t y_t + c_t x_t + r_t z_t) \tag{1}$$

subject to

$$
\begin{aligned}
x_t + y_t - y_{t-1} &= S_t \;\text{ for } t \geq 1, \; (y_0 = 0) && (2) \\
z_{t-1} + x_t - z_t &\leq D_t \text{ for } t \geq 1, \; (z_0 = 0) && (3) \\
x_t,\, y_t,\, z_t &\geq 0 \text{ for } t \geq 1 && (4)
\end{aligned}
$$

Note that, in a slightly non-standard fashion, a less-than-or-equal-to constraint is associated with demand. This, of course, could be changed. This form is consistent with the application to ground delay programs, we later describe. In what is probably the "classic" application of the model (see Section 4.5 of [4]), the state 1 commodity is crude oil and the state 2 commodity is refined oil. In this application one can view demand as the maximum amount of refined oil that can be sold during a period.

Note that the formulation of this model requires the estimation of a vector of future demand levels, $(D_1, ..., D_T)$. There will, of course, be uncertainty associated with such estimates. The generalization of this model, which we investigate, replaces the estimate of a single demand vector with a demand distribution. The demand distribution we employ involves a set of demand scenarios $\{(D_{1q}, ..., D_{Tq}) : \text{for } q = 1, ..., Q\}$ with associated probabilities, p_q for $q = 1, ..., Q$. In the model, the state 2 inventory levels will also vary with the scenario, so that we need additional variables:

z_{tq} = amount of commodity in state 2 held from period t to $t+1$ under demand scenario q

The associated stochastic programming model is illustrated in Figure 2 and is defined by:

$$\text{Min} \sum_{t=1}^{T} (a_t y_t + c_t x_t + \sum_{q=1}^{Q} p_q r_t z_{tq}) \qquad (5)$$

subject to

$$\begin{aligned} x_t + y_t - y_{t-1} &= S_t \text{ for } t \geq 1, (y_0 = 0) && (6) \\ z_{t-1\,q} + x_t - z_{tq} &\leq D_{tq} \text{ for } t \geq 1, q \geq 1, (z_{oq} = 0) && (7) \\ x_t, y_t, z_{tq} &\geq 0 && (8) \end{aligned}$$

We will show, in Section 2 of this paper, that the constraint matrix associated with this model is dual network. This implies that the model can be solved using network flow techniques or, alternatively, that integral solutions can be found using linear programming techniques. We also describe, in Section 1 of this paper, an application of this model to the ground holding problem in air traffic management. This application is significant in the sense that the problem is formulated in a way that is consistent with new decision support tools now being deployed by the FAA based on the Collaborative Decision Making paradigm.

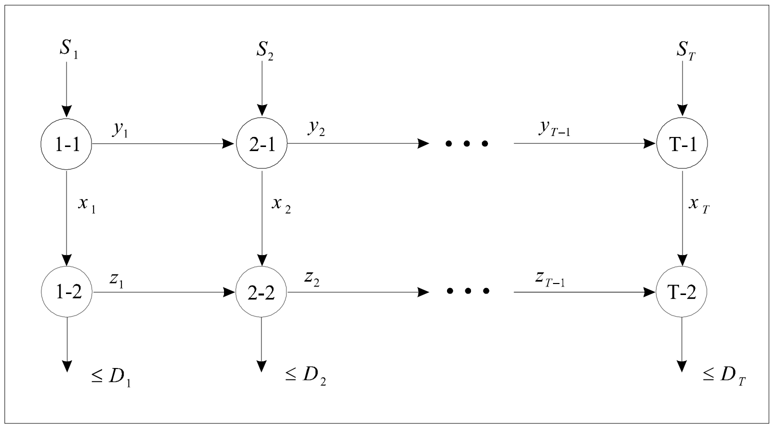

Figure 1: Network flow model for inventory with two states

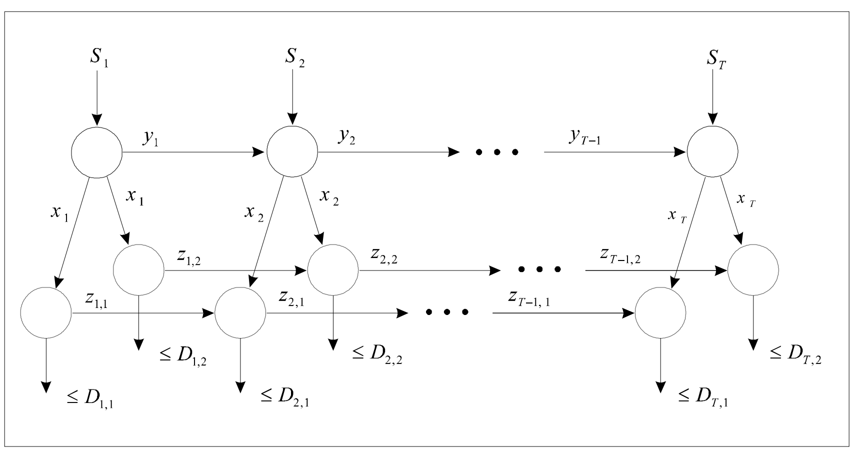

Figure 2: Stochastic flow model with two demand scenarios ($Q = 2$)

2 Application to Ground Holding Problem

Demand for runway operations (landings and take-offs) often exceeds available capacity for periods of several hours at major airports, usually due to unfavorable weather conditions. Ground Holding is the primary strategy used by The Federal Aviation Administration (FAA) to reduce arrival demand during such periods. The idea is to hold aircraft bound for a single destination on the ground at their origins for lengths of time sufficient to ensure that they will be able to land with little or no airborne delay. Ground holding is implemented over extended periods by means of a ground delay program (GDP). A GDP must be planned several hours in advance in order to capture aircraft originating at distant locations before they depart. The current practice is to convert a weather forecast into a profile of hourly airport acceptance rates (AARs), each of which is the number of aircraft that can be landed that hour. This partitions each hour into arrival slots of equal time lengths (varying landing time requirements are ignored) which are then rationed to flights. The method for assigning landing time slots to flights is now based on the Collaborative Decision Making (CDM) paradigm (see [6] or [12]). The CDM process involves substantial interaction between the airlines and the FAA. Controlled departure times are computed by subtracting scheduled or estimated travel times from controlled arrivals times. See [6] for a more in-depth description of ground delay program procedures.

Under current practice, GDPs are planned based on a single predicted AAR vector, $(\hat{A}_1, ..., \hat{A}_T)$, where $\hat{A}_1$ is the predicted arrival capacity during the 1st hour, $\hat{A}_2$, the predicted capacity during the 2nd hour, etc. Yet, this prediction depends on the weather forecasted several hours in advance. Clearly, there is a high degree of uncertainty associated with such predictions. As has been done previously in the literature (e.g., [8]), we characterize uncertainty using a discrete AAR distribution represented by a set of AAR vectors $\{(A_{1q}, ..., A_{Tq}) : \text{for } q = 1, ..., Q\}$ with associated probabilities, p_q for $q = 1, ..., Q$. Previous stochastic programming models took this input and then assigned landing time slots to individual flights. These types of models are invalid under CDM since individual time-slot-to-flight assignments result from a multi-stage process involving interaction between the airlines and the FAA.

The problem of interest in the CDM context is to "convert" the AAR distribution, $\{(A_{1q}, ..., A_{Tq}) : \text{for } q = 1, ..., Q\}$, into a single *Planned* AAR (PAAR) vector $(a_1, ..., a_T)$. We use lower case a here to emphasize that the

a_t are decision variables. The current practice of estimating the AAR at time t, $\hat{A}_t$, and then sending this number of flights to the destination airport, implicitly adopts the policy of setting the decision variable a_t equal to the most likely AAR value $\hat{A}_t$.

To understand the cost impacts of choosing a particular PAAR, consider the scenarios illustrated in Figures 3 and 4. In Figure 3, the actual AARs for periods 1 and 2 are 8 and 15, respectively (i.e. for the realized A_q vector, $A_{q1} = 8$ and $A_{q2} = 15$). The PAARs in periods 1 and 2 were set to 10 (i.e. $a_1 = a_2 = 10$). Under these conditions, 2 of the 10 flights scheduled to arrive during time interval 1 will be placed in an airborne queue and held over until time interval 2. There will then be 12 flights desiring to land during time interval 2, all of which will be able to land, since $A_{2q} = 15$. Given that ground holding is preferable to airborne holding (else there is no need for a GDP), it would have been better to have held two flights on the ground for one time unit so that they arrive at the terminal airspace at time $t = 2$. This is illustrated in Figure 4 where the single-unit ground delay of these two flights is represented by flow over a horizontal arc added on an upper level. Of course, since the realized AAR vector is not known in advance, it would not necessarily have been possible to predict that ground holding 2 flights would be the best strategy.

The key tradeoff illustrated is between assigned, deterministic ground delay and expected airborne delay. This tradeoff is driven by two parameters: g, the cost per time period of holding a flight on the ground and a, the cost per time period of airborne delay. One may assume $a > g$, otherwise, it would be cost effective to send all flights to the destination airport as soon as they were available.

It should now be clear that the problem of determining the PAAR vector can be solved by adding integer constraints to the stochastic model given in the introduction. A state 1 commodity is a flight on the ground and a state 2 commodity is a flight in the air. The problem input data and variables are as follows. The period t supply, S_t is the number of flights scheduled to arrive during time period t. The (maximum) period t demand under scenario q, D_{tq}, is the period t AAR under scenario q, A_{tq}. The per-period holding cost in state 1, a_t is set equal to g and the corresponding state 2 cost, r_t is set equal to a. The transformation cost, c_t, is set to 0. Two of the variable sets are relatively easy to interpret. The state 1 to state 2 transformation variable set, x, is the PAAR vector, i.e. $x_t = a_t$. The state 2 inventory variable z_{tq} is the number of flights held in the air from time period t to time period $t + 1$

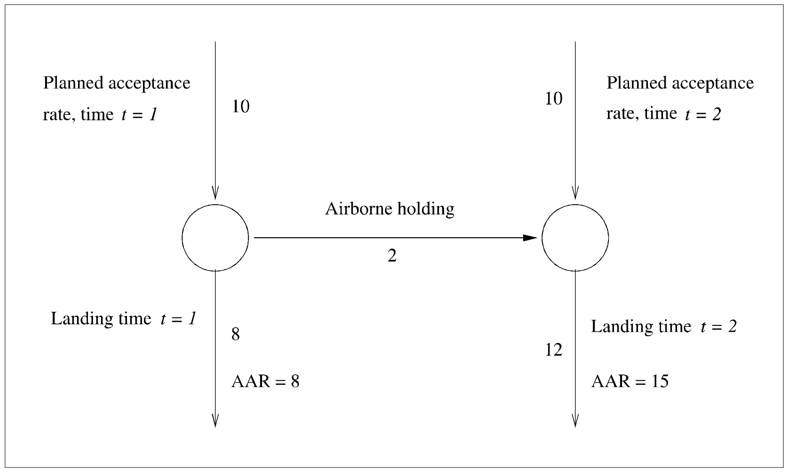

Figure 3: Airborne holding of two flights for one time period

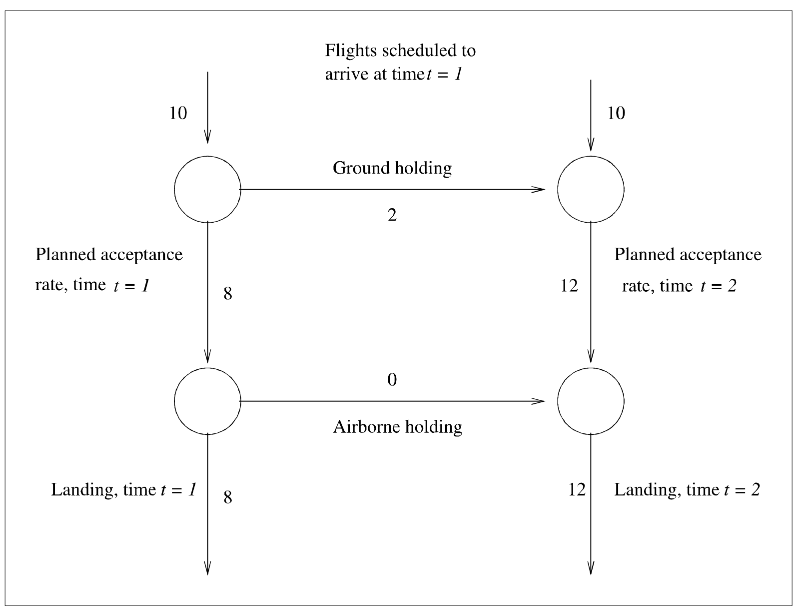

Figure 4: Ground holding of two flights in lieu of airborne holding

under scenario q. The interpretation of the y variable set requires discussion of a subtle aspect of the model. In a certain sense, the model does all time calculations relative to arrival times at the destination airport. Thus, y_t is the number of flights held on the ground from time period t to $t+1$. Time period t, here, however, is the time period in which the flight *would have arrived*. To determine when the flights are actually held on the ground, the individual flights must be chosen, the enroute time subtracted from the arrival time for each flight and finally the delay added to each flight's departure time.

A large body of work exists on deterministic versions of the ground-holding problem. See Terrab [10], Bertsimas and Vranas [2], Vranas, Bertsimas and Odoni [11], and Bertsimas and Stock [1]. Richetta and Odoni [8] introduce a stochastic ground holding model that employs a discrete capacity (AAR) distribution and the minimization of expected costs.

The model described here, which is based on the theses of Hoffman [5] and Rifkin [9], is distinguished in several ways. First, it outputs the aggregate number of flights that should be assigned controlled arrival times for each hour, rather than an individual controlled arrival time for each flight. This allows the assignment of controlled arrival times to individual flights to be done later by CDM-based rationing and equity schemes. Secondly, this model admits a formal proof that the linear programming (LP) relaxation of the integer program (IP) is guaranteed to yield integral solutions. Lastly, the aggregation of flights allows it to be solved more rapidly than the Richetta and Odoni model (aggregation in the Richetta and Odoni model was done strictly to reduce the run time). See [5] for a sensitivity analysis of the model.

3 Theoretical Results

In this section, we show that the stochastic ground holding problem can be solved in polynomial time. In particular, the main result is Theorem 2, which shows that SGHP, the integer program defined in the previous section by (6) - (10), is a dual network flow problem, though it fails to be a primal network flow problem. As immediate corollaries, the constraint matrix associated with this integer program is totally unimodular, and the LP relaxation yields integral solutions.

We say that a $(0, 1, -1)$ matrix is a *network matrix* if each column contains at most two nonzero entries and whenever a column contains two nonzero entries, they sum to zero. If N is a network matrix, then one can efficiently solve a linear program over N as a network flow problem, via known algorithms. Moreover, if N is transformed into a matrix M via simplex pivots, then M is a network matrix in hidden form and can also be solved as a linear program over M. For this reason, we can extend the definition of network matrices to include matrices such as M.

The characteristic feature of this model is that it is "almost" a network flow. Constraint sets (2), (3) and (7) have the structure of flow conservation constraints. However, as proven in Theorem 1, the presence of a common flow variable, x_t, in each of the Q constraints in (8) destroys the network structure except in the case of $Q = 1$. Figure 1 illustrates the model for the case $Q = 1$, where it is a network flow problem, and Figure 2 illustrates the model for the case $Q = 2$, where it is not.

Theorem 1 *Let (I, C) be the primal constraint matrix of SGHP, where I is the identity matrix that is induced by the addition of slack variables. Let Q be the number of scenarios. Whenever $Q \geq 2$, (I, C) fails to be a network matrix (either directly or in hidden form).*

Proof. *Since every submatrix of a network matrix is also a network matrix, it suffices to show that C contains a submatrix B that is not a network matrix. We form B by intersecting those rows corresponding to parameters $S_1, S_2, D_{1,1}, D_{2,1}, D_{1,2}, D_{2,2}$ and those columns corresponding to variables $y_1, z_{1,1}, z_{1,2}, x_1, x_2$. Since $Q \geq 2$, B is well defined and has the following form.*

$$B = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 \end{bmatrix}$$

By way of contradiction, suppose that B is a network matrix. Then there is a corresponding undirected graph G and a basis of edges, $\{e_1, e_2, ..., e_6\}$ identified by the identity columns in (I, C) that form a spanning tree of G (see [3] or [7]). Every column of B is the edge-path characteristic vector of a path in G with respect to the edges $\{e_1, e_2, ..., e_6\}$. The nonzero entries of B establish the following paths in G.

$P_1 = \{e_1, e_2\}$ $P_2 = \{e_3, e_4\}$ $P_3 = \{e_5, e_6\}$ $P_4 = \{e_1, e_3, e_5\}$ $P_5 = \{e_2, e_4, e_6\}$

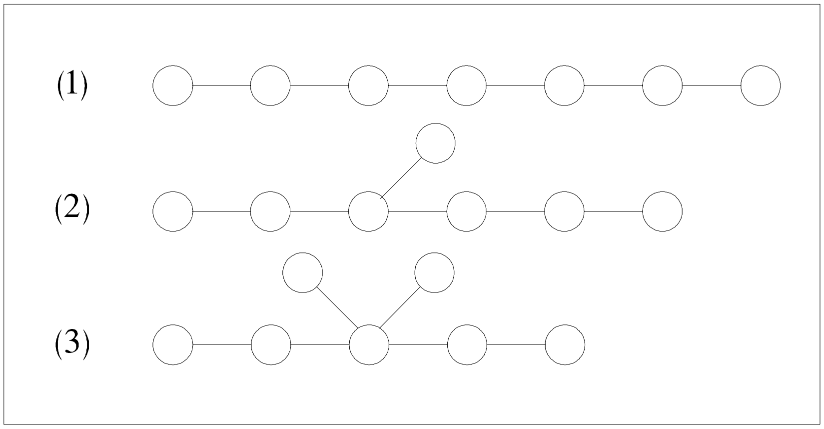

Figure 5: The three possible graph structures for G

Since G must be spanned by a tree with six edges, G contains exactly 7 nodes. Since P_4 and P_5 have four nodes (three edges) each, they must have exactly one node in common and G is isomorphic to one of the three graphs in Figure 5. But in each of these graphs, the left-most edge e_i is adjacent to exactly one edge, e_j. Edges e_i and e_j must lie together on either path P_4 or P_5, each of which indicate that i and j must have the same parity. But this contradicts the existence of paths P_1, P_2 and P_3, which indicate that every edge is adjacent to an edge with opposite parity. ■

Theorem 2 *Let C^T be the transpose of the constraint matrix C in $SGHP$ and let I be the identity matrix induced by adding slack variables to the dual of $SGHP$. Then $M = \left(I, C^T\right)$, the dual constraint matrix of $SGHP$, is a network matrix.*

Proof. *In order to show that M is a network matrix, it will suffice to show the existence of another $m \times (m+n)$ matrix N that can be transformed into M via simplex pivots. Let Q be the number of scenarios and let T be the number of time periods in SGHP. Let Δ and Δ' be the $(Q+2) \times (Q+2)$ matrices whose entries are defined via*

$$\Delta_{i\,j} = \begin{cases} 1,\ \textit{if}\ (i=1\ \textit{and}\ j=1)\ \textit{or}\ (i=2\ \textit{and}\ j=2) \\ -1,\ \textit{if}\ i-1=j \\ 0,\ \textit{otherwise.} \end{cases}$$

$$\Delta^{'}_{i\,j} = \begin{cases} -1, \text{ if } i = 1 \text{ and } j = Q+2 \\ 0, \text{ otherwise.} \end{cases}$$

Let θ be the $(Q+1) \times (Q+1)$ matrix whose entries are defined by

$$\theta_{i\,j} = \begin{cases} -1, \text{ if } i = 1 \text{ and } j = Q+2 \\ 0, \text{ otherwise.} \end{cases}$$

We define N to be the $T \times 2T$ block matrix below.

$$N = \begin{bmatrix} \Delta & \Delta^{'} & & & & \theta & -\theta & & & \\ & \Delta & \Delta^{'} & & & & \theta & -\theta & & \\ & & \cdots & \cdots & & & & \cdots & \cdots & \\ & & & \Delta & \Delta^{'} & & & & \theta & -\theta \\ & & & & \Delta & & & & & \theta \end{bmatrix}$$

For example, when $Q = 2$ and $T = 2$,

$$\Delta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad \Delta^{'} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Note that $\Delta, \Delta^{'}$ and θ are network matrices. By nature of their relative positions in N, N is also a network matrix. When simplex pivots are performed successively on the main diagonal elements of N, then N is transformed into M. See [5] or [9] for examples of the matrix M and the network graph underlying N. ■

We comment that Theorem 2 defines a transformation from N to M via matrix multiplication. That is, $NE = M$, where E represents the composition of necessary simplex pivots. Since E is invertible, Theorem 2 implicitly defines the matrix E^{-1} that transforms M into N via $N = ME^{-1}$.

Corollary 3 *The constraint matrix associated with SGHP is totally unimodular.*

Corollary 4 *The linear programming relaxation of SGHP yields integral solutions.*

Corollary 5 *The integer program SGHP can be solved to optimality in polynomial time.*

4 Implementation

Work is currently underway by the authors to incorporate this model into the CDM decision support tool, FSM (the Flight Schedule Monitor). FSM already provides the required demand estimates. Research is underway to estimate the AAR distributions. This research is based on an airport-specific statistical analysis of historical weather and GDP information. Given that the model operates at an aggregate level, it is probably appropriate to set the values of a and g according to industry wide standards. We remark that the optimal solution depends only on the *ratio* of these parameters $\frac{a}{g}$. An alternative interpretation of this cost ratio is a quantification of the FAA's willingness to trade ground delay for air delay, taking into account the trade-off between its desire to serve the industry efficiently and its operational and safety concerns. Under this interpretation, the cost ratio would be set by FAA policy, perhaps through historical documentation of practices.

5 Acknowledgments

This work was supported by the National Center of Excellence for Aviation Operations Research, under Federal Aviation Administration Research Grant Number 96-C-001 and contract number DFTA03-97-D00004. Any opinions expressed herein do not necessarily reflect those of the FAA or the U.S. Department of Transportation.

References

[1] Dimitri Bertsimas and Sarah Stock. The Air Traffic Flow Management Problem with Enroute Capacities. *Operations Research*, 46:406-422, 1998.

[2] Dimitri Bertsimas and Peter B. Vranas. Dynamic Ground-Holding Policies for a Network of Airports. *Transportation Science*, 28:275-291, 1995.

[3] Robert E. Bixby and William H. Cunningham. Converting Linear Programs to Network Problems. *Mathematics of Operations Research*, 5:321-357, 1980.

[4] F. Glover, D. Klingman, and N. Philips. *Network Models in Optimization and Their Applications in Practice*. John Wiley and Sons, Inc., New York, 1992.

[5] Robert Hoffman. *Integer Programming Models For Ground-Holding in Air Traffic Flow Management*. Ph.D. thesis, University of Maryland, 1997.

[6] Robert Hoffman, William Hall, Michael Ball, Amedeo Odoni, and Michael Wambsganss. Collaborative Decision Making in Air Traffic Flow Management. Technical Report RR-99-2, NEXTOR, UC Berkeley, 1999.

[7] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley and Sons, Inc., New York, 1988.

[8] Octavio Richetta and Amedeo R. Odoni. Solving Optimally the Static Ground-Holding Policy Problem in Air Traffic Control. *Transportation Science*, 27 #1, August 1993.

[9] Ryan M. Rifkin. The Static Stochastic Ground-Holding Problem. Master's thesis, Massachusetts Institute of Technology, 1998.

[10] Mostafa Terrab. *Ground Holding Strategies for Air Traffic Control*. Ph.D. thesis, Massachusetts Institute of Technology, 1990.

[11] Peter B. Vranas, Dimitri Bertsimas, and Amedeo R. Odoni. The Multi-Airport Ground-Holding Problem in Air Traffic Control. *Operations Research*, 42:249-261, 1994.

[12] Michael Wambsganss. Collaborative Decision Making Through Dynamic Information Transfer. *Air Traffic Control Quarterly*, 4:107-123, 1997.