A Comparison of Formulations for the Single-Airport Ground Holding Problem with Banking Constraints

Article in Operations Research · October 1998

DOI: 10.1287/opre.48.4.578.12417 · Source: CiteSeer

Robert Hoffman
Metron Aviation

TECHNICAL RESEARCH REPORT

A Comparison of Formulations for the Single-Airport Ground Holding Problem with Banking Constraints

by R. Hoffman, M. Ball

T.R. 98-44

A Comparison of Formulations for the Single-Airport Ground Holding Problem with Banking Constraints

by

Robert Hoffman
Applied Mathematics Program
and
Institute for Systems Research
University of Maryland
College Park, MD 20742
and
Michael O. Ball
College of Business and Management
and
Institute for Systems Research
University of Maryland,
College Park, MD 20742

October, 1997

ABSTRACT

Both the single-airport ground-holding problem (GH) and the multi-airport ground-holding problem can be extended by the addition of banking constraints to accommodate the hubbing operations of major airlines. These constraints enforce the desire of airlines to land certain groups of flights, called banks, within fixed time windows, thus preventing the propagation of delays throughout their entire operation. GH can be formulated as a transportation problem and readily solved. But in the presence of banking constraints, GH becomes a difficult integer programming problem. In this paper, we construct five different models of the single-airport ground holding problem with banking constraints (GHB). The models are evaluated both computationally and analytically. For two of the models, we show that the banking constraints induce facets of the convex hull of the set of integer solutions. In addition, we explore a linear transformation of variables and a branching technique.

1. Introduction

As of the mid-1980's, air traffic congestion in the Unites States has become an increasing problem, particularly at the major airports. Although air traffic demand is problematic, it is primarily comprised of scheduled flights, hence, it is generally predictable. Airport capacity, on the other hand, can change sharply and with little warning. Most of the airport capacity-demand inequities are the result of a sudden drop in arrival capacity rather than an unforeseen insurgence of arrival demand. Bad weather is the primary cause. Precipitation and icing can shut down runways altogether and aircraft must approach more slowly and cautiously when relying on instruments rather than on human vision. Special airport operations, visiting dignitaries and runway construction also contribute to reduced arrival capacities.

A capacity-demand inequity at an airport can lead to queuing of both departing aircraft and arriving aircraft. Queuing as a result of reduced arrival capacity is considered to be a more serious problem, though, because it forces the aircraft into airborne holding patterns, which is costly, dangerous and adds to the stress level of the air traffic controllers. In this paper, we address capacity-demand inequities in arrivals only.

The Air Traffic Control Systems Command Center (ATCSCC) monitors airports throughout the United States for capacity-demand inequities. Whenever it is predicted that the number of flights arriving at an airport within a 15-minute time interval will exceed the number of flights scheduled to land, the ATCSCC takes action. Short-term periods of capacity-demand inequities can be alleviated by airborne tactics such as re-routing and variations in airborne speed. Longer-term periods of capacity-demand inequities are met by the ATCSCC with ground-holding strategies in which aircraft are held at their departure gates in lieu of costly and dangerous airborne delay.

The primary tool of the ATCSCC for addressing arrival capacity-demand inequities is a ground delay program (GDP). In a GDP, each flight scheduled to arrive at an afflicted airport over a predetermined time period is held at its departure gate long enough to ensure that it will be able to land without delay. For instance, if flight *f* is scheduled to arrive at airport A at 12:00 and it is known that *f* will not be able to land until 12:30 due to limited arrival capacity at A, then *f* would be held at its departure gate for 30 minutes.

Currently, when the ATCSCC formulates a GDP, arrival slots are assigned on a 'first-scheduled, first-assigned' basis. That is, if flight *f* were originally scheduled to arrive before flight *g*, then *f* should arrive before *g* in the final slot assignments. However, the entire process of assigning allocating slots during a GDP has fallen under heavy scrutiny and is currently being revamped by a large-scale, cooperative effort between the FAA and the scheduled carriers, known as collaborative decision making (CDM). Under the proposed CDM procedure, (to be implemented in the fall of 1997), flights will initially be assigned to time slots on a first-scheduled, first-assigned first basis. Then, in an iterative exchange between the airlines and the ATCSCC, each airline will have the opportunity to reassign some of its flights to its allocated arrival slots, thus giving the airlines greater control over the economic impacts of a GDP.

An area of our future research is to understand the axioms imposed by these algorithms and to incorporate them into a single optimization model. For the purposes of this paper, we will view the CDM allocation procedure as a black box and model it as a generalized ground-holding problem (GH). In GH, a decision-maker is faced with reduced arrival capacity at an airport and must determine the appropriate amount of ground delay to assign to each incoming flight so as to minimize overall delay costs. This problem can be formulated as a *transportation problem*, as follows. We discretize the time horizon into time periods $t = 1,2,\ldots,T$. Each time period could represent, say, a 15-minute time interval. For each flight *f*, let a_f be the scheduled time of arrival. For each time period, *t*, let b_t to be the arrival acceptance rate (AAR) of the airport, i.e., the maximum number of flights that can be accepted by the airport during that time interval. We assume that b_t is known in advance for each time period *t*. (Strictly speaking, this last assumption does not hold in practice but the specialist must fix these numbers according to the current best estimate thus, for purposes of this formulation, we will assume that arrival capacity for each time period is deterministic and known in advance.

Each flight in a ground delay program is assigned a controlled time of arrival (CTA) and a controlled time of departure (CTD). Since en route travel times can be predicted with reasonable accuracy, both the CTD and the amount of assigned ground delay are easily computed once the CTA is fixed: the controlled time of departure (CTD) is simply the CTA minus the en route time and the ground delay is the CTD minus the scheduled arrival time. Thus, a feasible solution to the single-airport ground-holding problem can be derived once each flight has been assigned a CTA.

Let F be the set of incoming flights that require arrival slots. We define for each *f* and each *t*, a binary variable, X_{ft}, such that

$$X_{ft} = \begin{cases} 1, & \textit{if flight f is assigned to time interval t} \\ 0, & \textit{otherwise} \end{cases}.$$

Then we have the following integer program.

(GH) $$Min \sum_{f \in F} \sum_{t=1}^{T} C_f \left(t - a_f\right)^{\sigma} X_{ft} \tag{1.1}$$

subject to

(assignment) $$\sum_{t=1}^{T} X_{ft} = 1 \quad \textit{for all } f \tag{1.2}$$

(capacity) $$\sum_{f \in F} X_{ft} \leq b_t \quad \textit{for all } t \tag{1.3}$$

$$0 \leq X_{ft} \leq 1 \quad \textit{for all } t, \textit{ for all } f \tag{1.4}$$

$$X_{ft} \in \{0,1\} \quad \textit{for all } f, \textit{ for all } t \tag{1.5}$$

where,

b_t = arrival capacity of airport during time interval t

C_f = a constant peculiar to flight *f*

$\sigma > 1$ is a fixed parameter.

Constraint set (1.1) ensures that each flight *f* is assigned to exactly one time interval *t* while constraint set (1.2) ensures that the capacity of each time interval is not exceeded. The objective function reflects overall delay costs. The parameter $\sigma > 1$ is used for super-linear growth in the tardiness of a flight so that the model tends to favor a moderate amount of delay to each of two flights rather than the assignment of a small amount of delay to one and a large amount to the other.

GH was first systematically described by Odoni in [7]. Since that time, GHP has been treated on a stochastic level by Odoni, Andreatta and Richetta in [9] and [10]. Both GH and traffic flow management in general have been treated on a network-wide level (taking multiple airports and flight connectivity into account) in Attwool [3], Sokkapia [11], Andreatta and Romanin-Jacur [1], Wang [15] and by Vranas, et. al., in [13] and [14], and, more recently, by Bertsimas and Stock [5]. However, in this paper, we will restrict our attention to single-airport scenarios and the deterministic version of GH.

Since the LP relaxation of GH (GH_{LP}) is a transportation problem, LP solvers or specialized transportation codes can be applied to GH_{LP} to obtain the (integer) solution to GH.

For operational efficiency, most major airlines in the United States have selected at least one airport as a *hub* of its operation. The hub acts as a base of operation and a central point of transfer for passengers, thus simplifying the enormous scheduling problem that confronts the airline. The hub-and-spoke system allows an airline to pool at a central location those passengers with geographically diverse points of origin but a common destination (or the reverse). For instance, some of the passengers from flights A, B and C can be scheduled to transfer at the hub to a flight D with a destination common to all of them. But in order for this to work, the arrival of flights A, B and C need to be coordinated with the departure of D. Flights A, B and C form what is known as a *bank*, meaning, a group of flights whose arrival times must fall within a specified time window.

In the solution to GH, the assigned arrival times of the flights tend to spread out over time because the number of flights that can be accepted per time period is less than in the original schedule. Of course, this tends to spread out the arrival of flights within a bank as well, often beyond an acceptable level.

One can add *banking constraints* to the formulation of the GH to keep the flights of each bank temporally grouped. For each bank *b*, let Φ_b be the set of flights in bank *b* and let w_b be the width of *b*, meaning the maximum number of time intervals over which the flights of bank *b* are allowed to land. Note that the difference between the sums $\sum_{t=1}^{T} tX_{f\,t}$ and $\sum_{t=1}^{T} tX_{g\,t}$ is the difference between the arrival times of the flights *f* and *g*. Then the following constraint set, for instance, will ensure that the flights of *b* land in a time window of desired length.

Formulation 1: XTC (the time coefficient model)

$$\sum_{t=1}^{T} tX_{ft} - \sum_{t=1}^{T} tX_{gt} \leq w_b \quad for\, all\, b,\;\; for\, all\, (f,g) \in \Phi_b \times \Phi_b \tag{1.6}$$

By adding (1.6) to GH, we have a model (XTC) of GHB. Unlike GH, the LP relaxation of the GHB rarely yields optimal integer solutions. In this paper, we will be exploring alternate formulations of GHB. Alternate formulations can be derived by reformulating the constraints, selecting new variables, or augmenting the existing ones. Ideally, we would a formulation of GHB that can be solved quickly on a commercial solver such as CPLEX.

In sections 2 and 3 of this paper, we present several models of GHB and explain some of the intuition behind them. In section 4, we analyze the polyhedra induced by some of the more promising models and, in section 5, we test the computational performance of each model on both real and artificially constructed data sets. We summarize the paper in section 6.

We should mention that, although this paper is written in the context of air traffic management, the problem is in its most general form a job scheduling problem in which a number of sequenced jobs must be scheduled for processing subject to the constraint that only so many jobs can be processed in a given time period and the (banking) constraint that certain jobs must be processed within temporal proximity of each other.

2 Alternate Models of GHB

Formulation 2: XW (the Window model)

It seems, intuitively, that the solving of GHB would be greatly facilitated by advanced knowledge of the time window in which each bank will arrive in the optimal solution. Each such window can be uniquely identified by its first time interval (i.e., the one with the lowest index value, *t*). This is the earliest time interval to which any of the flights of bank *b* can be assigned. So, for each bank, *b*, we establish a set of binary "marker" variables as follows.

$$Z_t^b = \begin{cases} 1, & \textit{if } t \textit{ is the first time interval open to bank } b \\ 0, & \textit{otherwise.} \end{cases}$$

We can use the marker variables to write a constraint that says, "if *t* is the earliest time interval open to the flights of bank *b*, then the arrival time of flight *f* in bank *b* must be no later than w_b units after *t*". We need one such constraint for each flight in each bank.

$$Z_t^b - \sum_{s=t}^{t+w_b-1} X_{fs} \le 0 \quad \textit{for all } t, \ \textit{for all } b, \ \textit{for all } f \in \Phi_b \tag{2.1}$$

The following set of assignment constraints ensures that the first time interval open to each bank is unique.

$$\sum_{t=1}^{T} Z_t^b = 1 \quad \textit{for all } b \tag{2.2}$$

The model XW is obtained by adding constraint sets (2.1) and (2.2) to GH. This model yields at most one banking constraint of type (2.1) for each pair (*f*, *t*), where $f \in F$ and $t \in \{1, 2, \ldots, T\}$, and one banking constraint of type (2.2) for each *b*. Thus, the total number of banking constraints is O(nT), where n is the number of bank flights and T is the number of time intervals.

Formulation 3: XMM (the Monotone Markers model)

An alternate formulation of the window constraint (C3.XW) can be written by directly translating the statement "if flight f (in bank b) arrives in time interval t, then one of the w_b intervals prior to t must be marked as the first interval open to bank b". This is the converse of the statement that generated (2.1) in the model XW.

$$X_{ft} - \sum_{s=t-w_b}^{t-1} Z_s^b \leq 0 \quad \text{for all } t,\ \text{for all } b,\ \text{for all } f \in \Phi_b \tag{2.3}$$

Rather than mark the first time interval by $Z_t^b = 1$, and $Z_t^b = 0$ for all other time intervals (as in XW), we can mark all time intervals strictly preceding the start of the window by the assignment $Z_t^b = 1$ and all subsequent intervals by $Z_t^b = 0$. (In essence, we are transforming the marker variables into Bertsimas-Stock variables - see section 3 for an explanation of these variables and why they might help). Constraint set (2.4) excludes the possibility that both X_{ft} and Z_t^b are equal to one for a fixed t while constraint set (2.5) forces the marker variables to be monotonically non-increasing.

$$Z_t^b + X_{ft} \leq 1 \quad \text{for all } t,\ \text{for all } b, \text{for all } f \in \Phi_b \tag{2.4}$$

$$Z_t^b - Z_{t-1}^b \leq 0 \quad \text{for all } t, \text{for all } b \tag{2.5}$$

The model XMM is obtained by adding (2.3), (2.4) and (2.5) to GH. The number of banking constraints increases quadratically with the size of the problem and has an asymptotic bound of O(nT).

Formulation 4: XSS (the Double Sum model)

The following simple constraint states that if flight *f* arrives in time interval *t*, then flight *g* cannot arrive in time interval *s* and vice-versa.

$$X_{ft} + X_{gs} \le 1 \tag{2.6}$$

If we write one constraint of type (2.6) for each pair of bank flights *f* and *g* and for each pair of time intervals *t* and *s* such that $|t - s| > w_b$, then all the flights of bank *b* must arrive within a window of w_b units.

We can write a stronger version of (2.6), which states that if *f* lands in time interval *t* or earlier than *g* cannot land in time interval $t + w_b$ or later, as below.

$$\sum_{s=1}^{t} X_{fs} + \sum_{ss=t+w_b}^{T} X_{gss} \le 1 \qquad \text{for all } t,\ \text{for all} \left(f,g\right) \in \Phi_b \times \Phi_b \tag{2.7}$$

The final model, XSS, is obtained by adding (2.7) to GH. We extend the notion of "arrival" to fractional solutions by saying that if $X_{ft} > 0$, then *f* has *partially arrived* at time *t* and *f* has *fully arrived* at the earliest time interval *t* for which $\sum_{s=1}^{t} X_{fs} = 1$. For each bank *b*, let $A_b = \left\{X_{ft}\colon\ f \in b\right\}$. Then in any solution to the linear relaxation of the GHB, one can compute the minimum and maximum values of *t* for which at least one of the variables in A_b is non-zero. We define the *range of the bank* in a given solution to be the difference of those numbers.

The strength of our latest formulation, XSS, lies in its ability to keep this bank range as small as possible in the LP. That is, XSS screens out fractional solutions in which the range of the bank is large. As an example of a fractional solution that is feasible to constraints of the type (1.2)-(1.4) but not to (2.7), consider two flights, *f* and *g*, in bank *b*, with a specified bank width of w_b = 2 time intervals. The table below gives a feasible assignment for the variables X_{ft} and X_{gt} for the time intervals, *t* = 1, 2, …, 8.

$t =$	1	2	3	4	5	6	7	8
$X_{ft} =$	1/2	1/2	0	0	0	0	0	0
$X_{gt} =$	0	0	0	0	0	1/2	1/2	0

The model XSS has the undesirable feature that it produces a tremendous number of constraints for large problems. In fact, the number grows cubically with the size of the problem; it's asymptotic behavior is O(Tn^2). On the largest data set that we tested, 114,855 of the 115,174 constraints (i.e., 99.73%) were banking constraints. For problems of this size, even the compilation time of the C-program that writes the input for the solver CPLEX is significant: on the order of ten minutes. We now search for a model of equal strength that brings with it fewer constraints.

Formulation 5: XGF (the Ghost Flight model)

So far, in our formulations of banking constraints, we have made pairwise comparisons of the arrival times of the flights within a bank. But if we knew that, in every feasible solution to GHB, a "pilot" flight in the bank were going to arrive before the other flights in the bank, then we could compare the arrival of each bank flight to the pilot flight and cut down on the number of constraints by an order of magnitude.

There is no reason to believe, á priori, that every bank would naturally contain a pilot flight but we can add a *ghost flight* to each bank and write a constraint to enforce the arrival of the ghost flight before the other flights in the bank. For each bank b, we define a set of assignment variables, $\{ Z_t^b : t = 1, 2, \ldots , T\}$, to mark the (fictitious) arrival of the ghost flight. That is, $Z_t^b = 1$ if the ghost flight arrives at time t and $Z_t^b = 0$, otherwise. The following constraint set will ensure that the arrival of each ghost flight is unique.

$$\sum_{t=1}^{T} Z_t^b = 1 \quad for\, all\, b \tag{2.8}$$

For each flight f in bank b, we write a constraint of the type (2.9) to ensure that the ghost flight will arrive before flight f and a constraint of type (2.10) to prevent the flights of each bank b from arriving more than w_b units behind the bank's ghost flight.

$$\sum_{s=t}^{T} Z_s^b - \sum_{s=t}^{T} X_{fs} \leq 0 \quad \textit{for all } t,\ \textit{for all } b,\ \textit{for all } f \in \Phi_b \tag{2.9}$$

$$\sum_{s=1}^{t} Z_s^b + \sum_{ss=t+w_b}^{T} X_{f\,ss} \leq 1 \qquad \textit{for all } t,\ \textit{for all } f \in \Phi_b \tag{2.10}$$

The final model, XGF, is obtained by adding (2.8), (2.9) and (2.10) to GH. For every bank flight f and every time interval t, this model yields one banking constraint of the type (2.10) and one of the type (2.9). For every bank b and every time interval t, there is one constraint of the type (2.8). Thus, the total number of banking constraints produced by this model is O(nT), where n is the number of bank flights. Contrast this with O(n^2T) for model XSS.

In section 4, we will show that XSS and XGF are of equal strength, meaning that they optimal function value for the LP is the same for each model. Moreover, we will see that for both XSS and XGF, every banking constraint is a facet of the polyhedron formed by the set of integer solutions. This is most desirable because it greatly increases the chances of yielding an integer solution directly from the LP relaxation.

3 Variations on the Models

3.1 A Branching Technique:

Recall that several of our formulations employ marker (Z) variables. If the (binary) value of each marker variables is fixed, then each banking constraint reduces to a trivial statement or is redundant to a non-banking constraint. The subsequent LP relaxation is a transportation problem and will yield an integer solution. Thus, we obtain a valid formulation by restricting only the Z variables to be integer. The IP solvers will then only branch on the Z variables.

This branching technique was applied to XW, XMM, XSS and XGF. In Tables 1-7 (Appendix A), the reader will find rows marked "XWZ" and "XMMZ". These formulations are MIP (mixed integer programs) versions of XW and XMM, respectively, because the assignment variables (X_{ft}) have not been declared integer. Neither XSS nor XGF model names are suffixed with a "Z" because we solved these models only with this special branching technique. In section 5, we will analyze the benefits of the branching technique.

2. Bertsimas-Stock variables: A linear transformation

One can replace the standard assignment variables, with so-called *Bertsimas-Stock* (B-S) variables, defined by

$$W_{ft} = \begin{cases} 1, \text{ if } \textit{ flight } f \textit{ arrive by time } t \\ 0, \textit{otherwise} \end{cases}.$$

The assignment variables are defined so that for exactly one t, $X_{ft} = 1$. In contrast, the B-S variables are defined so that for <u>every</u> s greater than some t, $W_{fs} = 1$. Thus, every model that employs B-S variables requires the following set of monotonicity constraints.

$$W_{f\,t-1} - W_{f\,t} \leq 0 \quad \textit{for all } t, \textit{ for all } f \tag{3.1}$$

One can see that the standard variables are linearly related to the B-S variables via

$$X_{f\,t} = W_{f\,t} - W_{f\,t-1}. \tag{3.2}$$

In [5], Bertsimas-Stock versions of the multi-airport ground holding problem (MAGHP) performed quickly and often offered optimal integer solutions directly from the LP relaxation. According to Bertsimas and Stock, the B-S variables conveniently captured the connecting constraints of the MAGHP and were in many cases facetial in nature. Hoping for similar success with respect to our banking constraints, we applied the transformation (3.2) to models XSS and XGF to obtain WSS and WGF, respectively.

(WSS)

$$Min \sum_{f \in F} \sum_{t=1}^{T} C_f \left(t - a_f\right)^{\sigma} \left(W_{f\,t} - W_{f\,t-1}\right) \tag{3.3}$$

subject to

$$W_{f\,T} = 1,\, W_{f\,0} = 0 \quad for\ all\ f \tag{3.4}$$

$$\sum_{f \in F} \left(W_{f\,t} - W_{f\,t-1}\right) \le b_t \quad for\ all\ t \tag{3.5}$$

$$W_{f\,t-1} - W_{f\,t} \le 0 \quad for\ all\ t,\ for\ all\ f \tag{3.6}$$

$$W_{f\,t} - W_{g\ \ t+w_b-1} \le 0 \quad for\ all\ t,\ for\ all \left(f, g\right) \in \Phi_b \tag{3.7}$$

$$W_{f\,t} \in \left\{0,1\right\} \quad for\ all\ f,\ for\ all\ t \tag{3.8}$$

WGF is the same as WSS except that (i) we add one (ghost flight) binary variable set $\left\{W_t^b : t = 0, 1, \ldots, T\right\}$ for each bank b (ii) we add monotone constraint set

$$W_T^b = 1, \quad W_0^b = 0 \quad for\ all\, b \tag{3.9}$$

$$W_{t-1}^b - W_t^b \le 0 \quad for\ all\ t,\ for\ all\ f \tag{3.10}$$

and (iii) we replace (3.7) with the following two sets of banking constraints.

$$W_{f\,t} - W_t^b \leq 0 \quad for\,all\,t,\,for\,all\,b,\;for\,all\;f \in \Phi_b \qquad (3.11)$$

$$W_t^b - W_{f\,t+w_b} \leq 0 \quad for\,all\,t,\,for\,all\,b,\;for\,all\;f \in \Phi_b \qquad (3.12)$$

Since WSS and WGF are linear transformations of XW and XSS, they will yield the same objective function values (in the LP's) as their assignment variable counterparts. Moreover, since XSS and XGF are equivalent in the LP (see section 4 for proof), the LP optimal function value will be the same for all four models in every problem instance. This fact is confirmed empirically in Tables 1-7, Appendix A.

4. Polyhedral Results

The set of integer feasible solutions is the same for each of the models we have presented but the variations in the associated LP relaxations can drastically affect the performance of solution methods based on a branch-and-bound algorithm. Formulations are preferable for which the function value of the LP relaxation is close to the function value of the integer program. In this section, we investigate analytically the strength of the formulations XSS and XGF. We will employ the notation and basic results of polyhedral combinatorics, which can be found Nemhauser and Wolsey [6], and Pulleyblank [4]. We require the following additional notation**.**

GH = set of integer solutions to constraints (1.2),(1.3) and (4)

GHB_1 = set of integer solutions to constraints (1.2), (1.3), (4) and (2.7)

GHB_2 = set of integer solutions to constraints (1.2), (1.3), (4), (2.8), (2.9) and (2.10)

P^C = convex hull of P, where P is a given set of points in Euclidean Space

Then GH is the set of feasible solutions to the ground holding problem, GHB_1 is the set of feasible solutions to the double-sum formulation (XSS) and GHB_2 is the set of feasible solutions to the ghost flight formulation (XGF). We will show that, under mild assumptions, each of the banking constraints of the models XSS and XGF represents a facet of its respective polytope. We will show that the each capacity constraint (1.3) represents a facet of both GHB_1^C and GHB_2^C. Finally, we will show that XSS and XGF are equivalent in the strength of their LP relaxations. These results will be based upon the following assumptions.

Assumption 1. $b_T = F$. We assume that the capacity of the last time interval is the same as the number of flights. In would be true in practice to ensure feasible solutions. Our theoretical use of this assumption will be to construct feasible solutions in which an arbitrary number of flights has been assigned to the last time interval without affecting the optimal solution to the problem.

Assumption 2. $\sum_{i=t}^{i=t+w_b} b_i > \left|\Phi_b\right|$ for all *b* and all *t*. We assume that the capacities of the time intervals are sufficient to allow for the landing of any bank, *b*, over any block of w_b contiguous time intervals. Combined with assumption 1, this will allow us to generate a feasible solution in which bank *b* arrives in any chosen block of time intervals and all flights not in bank *b* arrive in time interval T. The full strength of this assumption is not required but the complexity of the weaker version would obscure the proofs.

Assumption 3. *For all t,* $b_t \geq 2$. In practice, a time interval would probably represent 10 minutes or more, hence, could accommodate at least two flights. The case in which $b_t < 2$ for some or all of the *t* might be interesting from a theoretical standpoint.

Assumption 4. We assume that for each flight *f*, $a_f = 1$. This means that flight *f* can be assigned to any one of the time intervals, t = 1, 2, …, T. This assumption eliminates pathological interactions between the flight arrival times and the bank structure and allows us to index the components of a feasible solution (vector) in the following uniform fashion.

$$X = (X_{11}\, X_{1\,2}\,,\ldots,\, X_{1\,T}\,|\, X_{2\,1} X_{2\,2}\,,\ldots,\, X_{2\,T}\,|\, X_{F\,1} X_{F\,2}\,,\ldots,\, X_{F\,T})\,.$$

For notational convenience, let $N = FT$ and $n = FT - F$. We begin by establishing the dimensionalities of the ambient polytopes.

Lemma 1: For each constraint C of the form (2.7), there are at least *n* affinely independent points of GHB_1^C that meet C at equality.

Proof: See Appendix B.

Lemma 2: $\text{Dim}\,(GHB_1^C) \geq n$.

Proof: See Appendix B.

Theorem 1: Dim(GHB_1^C) = Dim(GH)= *n*.

Proof: We have already shown that dim(GHB_1^C) ≥ n. Since $GHB_1^C \subseteq GH$, we have that dim(GH) ≥ dim(GHB_1^C). Because constraint set (1.2) generates F linearly independent equations, we have that dim(GH) ≤ $TF–F = n$, and the result follows.

•

When an instance of GHB is formulated by XGF rather than XSS, we have added one ghost flight to the problem for each bank. This increases the number of flights from F to $F + B$, where B is the number of banks and the dimension of the ambient Euclidean space for XGF has increased from $N = TF$ to $N^* = T(F + B)$, in the case of XGF. We would like to restrict our attention to the vector space R^{N^*} rather than alternate between R^N and R^{N^*}. So, let us assume that, in the formulation of XSS, we have added one ghost flight for each bank. This will not change the optimal solution since the variables corresponding to the ghost flights do not need to appear in any of the constraints. The purpose of this assumption is not only for notational simplification but also so that the solution vectors for XSS and XGF will have the same dimension and we can consider the feasibility of a single solution vector to either XSS or XGF and we can make use of our previous results. In particular, we can restate the conclusion of Theorem 1 to be that Dim(GHB_1^C) = Dim(GH)= n^*, where

$$n^* = T(F + B) - (F + B) = n + (TB - B) \ .$$

Theorem 2: Dim(GHB_2^C) = n^*, where $n^* = n + (TB - B)$.

Proof: Let Ω be the set of all solutions, X and Y, constructed in Algorithm 1. Each solution (vector) in Ω was constructed so that flight 1 lands before all other flights in bank b. Under the assumption that flight 1 is the ghost flight of bank b, each solution in Ω becomes feasible to GHB_2^C . As in Lemma 1, the vectors in Ω can be linearly combined to yield a set, Ω*, of n linearly (affinely) independent solutions to GHB_2^C . As in the proof of Lemma 2, one more linearly independent vector, U, may be added to Ω* to bring the total number to ($n^* + 1$). U is formed by setting $U = (Y - X)$, where Y and X are the integer solutions to GHB_2^C , described below.

Let $k = (w_b + t - 1)$.

In block 1: $Y_{1,k} = X_{1,k} = 1$ all other components are zero

In block 2: $Y_{2,k} = 1$ $X_{2,k+1} = 1$ all other components are zero

In block m (m ≠1, 2): $Y_{m,n} = X_{m,n}$ for all n. Set these binary components in any feasible manner.

This shows that $\dim(GHB_2^C) \geq n^*$. From Theorem 1, we know that $\dim(GH_1) = n^*$ and since $GHB_2^C \subseteq GH_1$, we conclude that $\dim(GHB_2^C) = n^*$.

•

The following lemma is used to establish that the banking constraints from model XSS induce facets.

Lemma 3: For every constraint C of the form (2.7), there is an integer point, $X \in GH$, that satisfies every constraint of the form (2.7) except C.

Proof: Let constraint C be given. This fixes bank flights f and g and a time interval, $t > w_b$, where $f, g \in \Phi_b$. For notational ease, let $w = w_b$ and let us drop the subscripts f and g from the assignment variables $X_{f\,t}$ and $X_{g\,t}$ so we can refer to the variables as X_t and Y_t, respectively. Also, we will assume that both flights are scheduled to arrive in the first time interval so that X_t and Y_t are defined for all t. Then the constraint C is given by

$$\sum_{s=1}^{t} X_s + \sum_{s=t+w}^{T} Y_s \leq 1. \tag{4.1}$$

Let S_1 be any solution that assigns $X_t = 1$ and $Y_{t+w} = 1$. Since $1 + 1 > 1$, S_1 violates constraint (4.1) . We will show that S_1 satisfies every other constraint of the form (2.7). Only certain of these constraints apply to the flights f and g and they come in two forms:

$$\sum_{i=1}^{\tau} X_i + \sum_{i=\tau+w}^{T} \mathrm{Y_i} \leq 1 \tag{4.2}$$

or

$$\sum_{i=1}^{\tau} Y_i + \sum_{i=\tau+w}^{T} X_i \leq 1 . \tag{4.3}$$

Since each summation in (4.2) and (4.3) is bounded between one and zero, it will suffice to show that exactly one of the two summations is zero. The four cases appear in Appendix B.

•

Theorem 3: Every banking constraint of the form (2.7) represents a facet of GHB_1^C and no two such constraints represent the same facet.

Proof: Fix a banking constraint, C, and let *F* be the face represented by C. Theorem 1 shows that there are n^* linearly independent (affinely independent) integer vectors of GHB_1^C that meet C at equality. Thus, dim(F) ≥ n^*. We know that dim(GHB_1^C) = n^*. Let *H* be the hyperplane represented by C. *H* has dimension greater than n^*, so we must consider the possibility that dim(F) = n^*. Let $GHB_1^C *$ be the polytope that results when constraint C is relaxed from GHB_1^C. By Theorem 1, and the fact that $GHB_1^C \subset GHB_1^C * \subset GH_2$ we know that dim($GHB_1^C *$) = n^*. Now dim(F) = n^* only if all of $GHB_1^C *$ lies on *H*. But Lemma 3 shows that a (unique) point of $GHB_1^C *$ is eliminated by this hyperplane. Thus, dim(F) < n^*. In all, dim(F) ≤ $n^* - 1$ and dim(F) ≥ $n^* - 1$, so dim (F) = ($n^* - 1$), and *F* is a facet of GHB_1^C, by definition. It follows from the uniqueness of the point in Lemma 3 that no two such constraints represent the same facet.

•

Theorem 4: Every banking constraint of the form (2.9) represents a facet of GHB_2^C and no two such constraints represent the same facet.

Proof: See Appendix B.

•

Theorem 5: Every banking constraint of the form (2.10) represents of a facet of GHB_2^C and no two such constraints represent the same facet.

Proof: Note that every facet of GHB_1^C is also a facet of GHB_2^C. (Recall that we have assumed the existence of ghost flights in the model XSS, so this statement is well defined.) Every ghost-flight constraint of the form (2.10) is a double-sum constraint of the form (2.7). We have already shown that every constraint of the form (2.7) is a facet of GHB_1^C and that the representation is unique.

•

Let F_t be the face of GHB_1^C (or GHB_2^C) represented by the capacity constraint corresponding to t. The conditions that are both necessary and sufficient for F_t to be a facet are extremely complex and peculiar to the problem instance. As we will see in the next theorem, a condition sufficient for F_t to be a facet is that there should be at least one solution feasible to all constraints except the capacity constraint. Since GHB is usually being solved under reduced capacity, it would not be hard to construct such a solution. For instance, if flights, $f_1, f_2, \ldots, f_{10}$ are scheduled to arrive in time interval t, and if the capacity of time interval t has been cut to, say, b_t = 7 flights, then one could assign $f_1, f_2, \ldots, f_7$ to time interval t and all other flights to time interval T. This type of construction would fail for an early time interval for which there are not enough flights to be assigned to it or when there is a bad interaction between bank flights and non-bank flights. For instance, suppose that the only way to fill the capacity of time interval t is to assign a particular flight, f, to time interval t. Then for every feasible solution, X, we have the implied equation, $X_{f\,t} = 1$. Since the variables over block f must sum to one, $X_{f,j} = 0$ for each $j \neq t$. This means that F_t has lost T dimensions, dim(F_t) < (n* (1.2), and F_t cannot be a facet of GHB_1^C (nor of GHB_2^C). But we consider this last scenario to be pathological. The hypothesis of the following theorem would most likely be true in practice.

Theorem 6: Let F_t be the face of GHB_1^C (or GHB_2^C) represented by the capacity constraint corresponding to time interval t. Then for each $t \neq T$, F_t is a facet of GHB_1^C (or GHB_2^C), provided that there is a set of b_t + 1 non-bank flights that can be assigned to t.

Proof: One can construct, for each $t \neq T$, a set Ω of n*-many linearly independent vectors such that each vector in Ω is a linear combination of vectors from F_t (see Appendix B for details of the construction). Therefore, F_t must contain n* linearly independent vectors. Since linearly independent vectors are affinely independent, it follows that dim(F_t) ≥ (n* (1.2). Recall that dim(GHB_1^C) = n* = dim(GHB_2^C). Since F_t is contained in GHB_1^C (and GHB_2^C), we have that F_t <= n*. Under the assumption that at least $b_t + 1$ flights can be assigned to t, there is at least one feasible solution that does not meet the capacity constraint at equality, hence, does not lie on F_t. Therefore, F_t is a proper subset of GHB_1^C (and GHB_2^C) and we can rule out the possibility that dim(F_t) = n*. It follows that dim(F_t) ≥ (n* (1.2) and so F_t is a facet by definition.

•

By using a polyhedral projection (see [3] and [6] for background), we will show that XSS and XGF are equivalent in strength. Let P_1 be a polyhedron defined over variable set x and let P_2 be a polyhedron defined over variable set (x, z). Then P_1 is the projection of P_2 onto x is if

$$P_1 = \{x\text{: there exists a } z \text{ with } (x, z) \in P_2\}.$$

Theorem 7: Let P_1 be the set of feasible solutions to the LP relaxation of XSS and let P_2 be the set of feasible solutions to the LP relaxation of XGF. Then P_1 is the projection of P_2 onto the variable x.

Proof: It will suffice to show that

(i) whenever $(x,z) \in P_2$, $x \in P_1$.

and (ii) whenever $x \in P_1$, there is a z such that $(x,z) \in P_2$

Proof of (i): Let $y = (x,z) \in P_2$. Fix time interval t and flights f and g in bank b. Because y satisfies every constraint of the form (2.9), we have that

$$\sum_{s=t+1}^{T} Z_s^b - \sum_{s=t+1}^{T} X_{fs} \leq 0 \quad . \tag{4.4}$$

The equalities below follow from (1.2) and (2.8), respectively.

$$\sum_{s=t+1}^{T} X_{f\,s} = 1 - \sum_{s=1}^{t} X_{f\,s} \tag{4.5}$$

$$\sum_{s=t+1}^{T} Z_s^b = 1 - \sum_{s=1}^{t} Z_s^b \tag{4.6}$$

By substituting (4.5) and (4.6) into (4.4), we obtain

$$\sum_{s=1}^{t} X_{f\,s} \ \le \ \sum_{s=1}^{t} Z_s^b \ . \tag{4.7}$$

For an arbitrary flight g in bank b, we add to a sum to each side of (4.7), to obtain the following inequality.

$$\sum_{s=1}^{t} X_{f\,s} + \sum_{ss=t+w_b}^{T} X_{g\ ss} \ \le \ \sum_{s=1}^{t} Z_s^b + \sum_{ss=t+w_b}^{T} X_{g\ ss} \tag{4.8}$$

Since y satisfies every constraint of the form (2.10), the right-hand side of (4.8), hence, the left-hand side of (32) is less than or equal to one. We have shown that, for an arbitrary time interval and pair of bank flights, the corresponding constraint of the form (2.9) is satisfied by x. The fact that x satisfies (1.2), (1.3) and (4.8) is trivial. Therefore, $x \in P_1$.

Proof of (ii): Let $x \in P_1$. For each bank b and each time interval t, we define

$B_t = \underset{f \in \Phi_b}{MAX} \sum_{i=1}^{t} X_{f\,i}$. For each bank b, we recursively define

$$Z_t^b = \begin{cases} B_t \ , & if\ t = 1 \\ B_t - \sum_{i=1}^{t-1} Z_i^b \ , & otherwise \end{cases} \tag{4.9}$$

Let z be the vector whose components are comprised of the variables defined in (4.9). We will show that (x, z) is in P_2 . By definition Z_t^b, we have that $\sum_{i=1}^{t} Z_i^b = B_t$. Since $0 \le B_t \le 1$ for each t, we have that $0 \le \sum_{i=1}^{t} Z_i^b \le 1$ for each t. Now whenever $t < \tau$, $B_t \le B_\tau$, so $\sum_{i=1}^{t} Z_i^b$ is non-decreasing, as t increases. Thus, for each t and b, Z_t^b is nonnegative and every constraint of the form $Z_t^b \ge 0$ is satisfied. The feasibility of x to XSS implies that $\sum_{s=1}^{T} X_{fs} = 1$ for every bank flight f and, in particular, $B_T = 1$. Since $\sum_{i=1}^{T} Z_i^b = B_T$, every constraint of the form (2.8) is satisfied for every bank b. These same constraints imply that for every $t > 1$ and every bank b,

$$1 - \sum_{s=t}^{T} Z_s^b = \sum_{s=1}^{t-1} Z_s^b \ . \tag{4.10}$$

Note that by definition z and B_{t-1}, $\sum_{s=1}^{t-1} X_{fs} = \sum_{s=1}^{t-1} Z_s^b = B_{t-1}$ for some flight f in bank b with the property that $\sum_{s=1}^{t-1} X_{fs} \ge \sum_{s=1}^{t-1} X_{gs}$ for every g in bank b. Thus, for every g,

$$\sum_{s=1}^{t-1} X_{gs} \le 1 - \sum_{s=t}^{T} Z_s^b \ . \tag{4.11}$$

By substituting $\sum_{s=1}^{t-1} X_{gs} = 1 - \sum_{s=t}^{T} X_{gs}$ into (4.11), we obtain the following constraint for every bank flight, g, and every time interval, $t > 1$.

$$\sum_{s=t}^{T} Z_s^b - \sum_{s=t}^{T} X_{g\,s} \leq 0 \tag{4.12}$$

In the event that $t = 1$, each of the summations in (4.12) is equal to one and the validity of the inequality is trivial. We have shown that *(x, z)* satisfies every constraint of the form (2.9).

Lastly, to show that *(x, z)* satisfies every constraint of the form (2.10), fix t and bank b. Let f be the flight corresponding to the maximum sum in the definition of B_t. For every $g \in \Phi_b$, there is a constraint of the following form (2.9) that is satisfied by x. That is,

$$\sum_{i=1}^{t} X_{f\,i} + \sum_{i=t+w_b}^{T} X_{g\,i} \leq 1\,. \tag{4.13}$$

By substituting $\sum_{i=1}^{t} Z_i^b \;=\; B_t \;= \sum_{i=1}^{t} X_{f\,i}$ in for the left-hand sum in (4.13), we see that *(x, z)* satisfies constraint (2.10) for an arbitrary t and flight f in an arbitrary bank b. Thus, *(x, z)* satisfies every constraint of the form (2.10) and $(x, z) \in P_2$, as desired.

•

Corollary 1: The LP relaxations to XSS and XGF have the same optimal objective function values.

Proof: Note that none of the auxiliary variables (Z_{bt}) appear in the objective function for XGF and that the objective functions for XSS and XGF are the same. The result follows from the preceding theorem.

•

5.1 The Data

We used five data sets to test the performance of the various formulations of GHB. Each data set was comprised of a set of flights, a collection of banks (subsets of the set of flights), the scheduled arrival times of the flights, and the capacities of the flights (i.e., the number of passengers that could be carried). The capacities were used to compute the weight of the flight in the objective function.

Data Sets 1 - 4 were constructed with a fictitious airport in mind with an arrival capacity of about one flight per minute. The total number of flights in each data set varied from 25 to 129 and the time horizon varied from 30 minutes to just over two hours. The arrival capacity is typical of a large metropolitan airport but the time horizons represent a relatively small slice of time. The time horizons were kept short to be sure that the problems could be solved in a reasonable amount of time. A more realistic time horizon would be on the order of 4-6 hours (as in data set 5) which would imply several hundred flights. Each problem instance was solved with a reduced arrival capacity of one-half the original arrival capacity (i.e., one flight per every two minutes).

The number of banks per data set was varied from one to seven, each bank consisting of eight to ten flights. In practice, this would be a small or medium-sized bank. The banks were scheduled to land over one to three time intervals. Since the time horizon was divided up into ten minute intervals, this translates to 10-30 minutes per bank. The bank densities (percentage of total flights that were bank flights) ranged from 8.9% to 45%.

We found that when a given data set (1- 4) is solved without banking constraints, each bank would tend to spread over about four time intervals (at ten minutes per time interval, that's a total of forty minutes). So, the bank spans were set at three time intervals (that's thirty minutes total) in order to keep the banking constraints active.

Data Set 5 was actual flight data taken over an eight-hour period at Chicago O'Hare Airport on February 12, 1993. By convention, GDP's are formulated and run over a four hour period so this data set represents a large instance of GHB. We solved the data set over the full eight hour period (13:00 - 20:59, data 5C) but not all the models were able to solve a problem this size, so we generated smaller data sets of four hours (13:00 -

16:59, data set 5A) and six hours (13:00 - 18:59, data set 5B) in order to test fully the performance of each model on real data.

Each problem instance was solved using CPLEX 3.0 on a SPARC 10 work station both as an LP relaxation and as an integer program (IP). We found little or no improvement in performance using the customization settings provided in CPLEX, so we stayed with the default settings.

With respect to the LPR, we were looking for

- *High optimal function values*
- *Low run times, and low iterations of the algorithm*

With respect to the IP, we were looking for

- *Ability to solve the IP within a node limit of 20,000*
- *Low run times, low number of iterations and low iterations of the algorithm*

The computational results are tabulated in Appendix A, Tables 1-7. In each data set, the delay constant for flight *f*, C_f, was set to one-tenth the passenger capacity of the aircraft. The time intervals are ten minutes each, so the function value units are roughly passenger-delay minutes. We say "roughly", because the delay cost of a flight grows exponentially with its tardiness.

5.2 The Findings

The *value gap* of a formulation is the percent by which the LPR optimal value varies from the IP optimal value. A lower value gap indicates a stronger model. In this respect, XGF proved to be the best of the five models. XSS, WSS and WGF will have the same performance relative to this metric since they have equivalent LP's. XGF yielded the lowest value gap in every data set. The value gap for XGF was never more than 2.32% and fell to zero in three of the data sets (1, 5A and 5B), indicating that the optimal integer solution was obtained directly from the LPR. We believe that the LP strength of the XGF model is due to the fact that each of its banking constraints represents a facet of the convex hull of the set of integer solutions.

Note that for each data set, XGF (but not necessarily XSS, WSS and WGF) solved the IP to integer optimality in very few nodes of the branch-and-bound algorithm (the most was 24 nodes for data set 4).

The run times for XGF (on the IP) varied from fractions of a second to just over 25 minutes (in data set 5B). GDP's are typically formulated a few hours in advance. The specialist would need time to review an optimal solution to GHB before making a final decision, so, in practice, the solution times that XGF displayed would most likely be acceptable.

The outstanding IP performance of XGF comes partly from its LP strength but also from the fact that we greatly reduced the number of nodes required in the branch-and-bound algorithm by the branching only on the Z-variables. Recall from section 3 that this branching technique was applied not only to XGF but to the other models that use marker variables (to mark the time window in which a bank lands): XWZ and XMMZ. In the tables, the formulations XWZ, XMMZ are the same as XW and XMM, respectively, but the IP was solved by branching only on the marker (Z) variables. Of course, the LP performances for XW and XWZ are the same (likewise, for XMM and XMMZ). However, the "Z" versions of these models vastly outperformed their counterparts in IP performance. For instance, the number of nodes that XWZ required to solve data set 4 was 16 nodes compared to 20,000 for XW.

This difference is so marked that we consider the establishment of marker variables and subsequent branching to be crucial toward solving in real time medium or large instances of GHB (or any such assignment problem with banking constraints).

At the lowest end of the performance spectrum lies the model XMM which, in every problem instance, ranked last in LP strength (high value gap), run time (both LP and IP) and number of nodes explored in the branch-and-bound algorithm. XMM was able to solve only the smallest of problems to integer optimality in the allotted thresholds of three hours and 20,000 nodes (data sets 1 and 2, which had only 25 flights and less than 25 time intervals).

5.3 Bertsimas-Stock Performance

We did not expect the B-S versions to find integer optimal solutions to the LP relaxations in cases where the standard versions did not since the LP are equivalent. Thus, differences will be related to LP solution times and branching issues.

In general, the B-S versions required more iterations to solve as an LP relaxation - often a full order of magnitude more than their standard counterparts. For instance, WGF required 1683 iterations to solve data set 3 (see Table 3, Appendix A) while XGF took only 657. The run times were not so widely different but the standard assignment variable models still outperformed the B-S versions.

For all but the smallest of data sets (i.e., more than 25 flights) the B-S models were outperformed by the standard assignment variable models. One possible reason for the poor performance of the B-S models relates to the replacement of non-negativity constraints with monotonicity constraints (essentially, there is an additional constraint for every variable). This would cause the simplex algorithm to spend significantly more time finding inverses of matrices, thus driving up the LP run times.

We conjectured that the B-S performance would become comparable to the standard versions if the problem had fewer variables. One way to cut down on the number of variables is to limit the amount of delay that could be assigned to any given flight. For instance, if a flight f were scheduled to arrive in the first time interval and there were a total of 25 time intervals, then with a 10 time period limit on the tardiness of each flight, one would need variables W_{ft} for $t = 1, 2, \ldots, 10$ rather than for $t = 1, 2, .., 25$. This type of limitation would be done in practice anyway since a flight is effectively canceled if it is severely delayed.

In order to test this hypothesis, we solved the LP relaxation of model WGF on data sets 4 and 5A, before and after upper bounds of 5 time units and 6 time units, respectively. The runtime of WGF dropped by about 61-72% while the number of iterations dropped by about 22-44% (see Table 8). However, we found comparable savings in run time and iterations (see Table 8) for XGF. The imposed bound did not close the performance gap between the two models.

A very significant property of the B-S models is that very simple constraints tend to represent facets. Recall that every banking constraint of XSS and XGF represented a facet of the convex hull of integer solutions. Since WSS and WGF are linear transformations of XSS and XGF, respectively, the banking constraints of WSS and WGF also represent facets for their respective polytopes. Note that these constraints involve only two variables.

5.4 Some Highlights of the Experiments

For data set 5, it took XGF just over 25 minutes to solve the six-hour time period (13:00 - 18:59, see data set 5B) whereas it took only 20 minutes to solve the eight-hour period (13:00 - 20:59, see data set 5C). One would think that it would take more time to solve an extension of a problem. We conjecture that the six-hour problem is equally difficult to solve because most of the bank flights are grouped in the first six hours of the eight-hour time period. We further conjecture that the node selection in the branch-and-bound algorithm may have been less fortunate in the six-hour case.

XTC turned out a surprisingly good performance on data set 5. Although its LP strength is less than that of XGF (or XSS), it solved data sets 5A and 5B in much less CPU time than XGF - sometimes an order of magnitude less. XTC required 1368 nodes of the branch-and-bound algorithm to solve data set 5C compared to only 3 for XGF and yet the solution times were comparable (around 20 minutes). This is because XTC was able to solve each iteration of the LP in much less time than XGF. This demonstrates that the strongest model (in LP strength) is not always the quickest way to solve an IP.

As one would expect, the length of time required to solve the LP and the IP grows with the time horizon and number of flights. All of the models were able to solve the small data sets (1 and 2) in less than a few seconds while on the larger data sets (5A, 5B, 5C) several of the models could not solve the problem in the (arbitrary) three-hour time limit. The relationship between size and run time is not strict, however. Data set 4 is smaller than data set 5A (120 flights versus 280 flights) and yet most models (XW and XMM in particular) had far more trouble solving data set 4. This might be because data set 4 had four banks whereas data set 5A had only two.

6. Closing Remarks

The single-airport ground-holding problem (GH) is a resource allocation problem in which each flight bound for an airport suffering reduced arrival capacity must be assigned to an arrival slot. We have explored various ways to add banking constraints to the single-airport ground-holding problem to enforce the temporal grouping of certain collections of flights known as banks. In all, we developed five basic models of the ground-holding problem with banking constraints. We showed analytically that two of these models, XSS and XGF, are equivalent in LP strength and that the banking constraints induce facets.

We tested the computational performance of the models on both real and constructed data sets. By branching on marker variables employed in several of the models, we obtained dramatic savings in obtaining integer solutions. The model XGF proved to be superior in every aspect of our computational testing. XGF is a powerful formulation of GHB that would perform well on real-world instances of the problem.

The computational performances of the Bertsimas-Stock versions of the models were disappointing. The primary reason was that the LP relaxations took much longer to solve. We managed to improve their performances by restricting the assignment of flights to excessively late time intervals. Even still, the models employing standard assignment variables prevailed in computational performance.

One way in which our work here could be extended is to reexamine the axiom that each bank must arrive strictly within its specified time band. A more realistic model might allow for the temporal expansion of a bank of flights beyond the desired parameter but at a penalty reflected in the objective function. The challenge there would be to find a concise mathematical representation of the expansion. Another option would be to allow the model to exempt (if necessary) one or two flights of each bank from the banking constraints. This might be a desirable trade-off for the reduction of overall delay costs. We note that, when applied in practice, our model would be embedded within a decision support tool such as the flight schedule monitor (FSM) currently used by CDM. By appropriate iterative use, solutions similar to those associated with these other models could be obtained.

Appendix A

Proof of Lemma 1:

The algorithm below produces *n*-many linearly independent, linear combinations of the vectors is *S*, where $n = (FT - F)$ and *S* is the set of integer solutions in GHB_1^C.

<u>Algorithm 1</u>

Note: Let $w = w_b$, for ease of notation.

STEP 1:

For $j = 1, 2, ..., T$

Set vector Y via:

Block 1: $Y_{1,j} = 1$ $Y_{1,k} = 0,\ k \neq j$

if $1 \leq j \leq (T - w+1)$

Block 2: $Y_{2,\ j+w-1} = 1$ $Y_{2,k} = 0,\ k \neq j + w$

else

Block 2: $Y_{2,\ T} = 1$ $Y_{2,k} = 0,\ k \neq T$

end if

Block p: (Set these components in any feasible manner)

$(p \neq 1, 2)$

Output row vector $U = Y$ (Note: $U_{1,j} = 1,\quad U_{1,k} = 0$ for all $k < j$)

end for

STEP 2:

For $j = 1, 2, ..., (T - 2)$

Set vectors X and Y via:

if $1 \leq j \leq (w(1.3)$

Block 1: $Y_{1,1} = 1$ $X_{1,1} = 1$ $Y_{1,k} = X_{1,k} = 0, k \neq j$

Block 2: $Y_{2,\ j} = 1$ $X_{2,j+1} = 1$ $Y_{2,k} = 0$ for $k \neq j$ and $X_{2,k} = 0$ for $k \neq j+1$

if $(w_b\,(1.2) \leq j \leq (w\,(1.2) + t -1$

$u = (j - w + 2)$

Block 1: $Y_{1,u} = 1$ $X_{1,u} = 1$ $Y_{1,k} = X_{1,k} = 0, k \neq u$

Block 2: $Y_{2,\ j} = 1$ $X_{2,j+1} = 1$ $Y_{2,k} = 0$ for $k \neq j$ and $X_{2,k} = 0$ for $k \neq j+1$

if $(w+ t\,(1.2) \leq j \leq (T - 2)$

Block 1: $Y_{1,j+1} = 1$ $X_{1,j+1} = 1$ $Y_{1,k} = X_{1,k} = 0, k \neq j+1$

Block 2: $Y_{2, j+1} = 1$ $X_{2, j+2} = 1$ $Y_{2, k} = 0$ for $k \neq j+1$ and $X_{2, k} = 0$ for $k \neq j+2$

end if

Block p: $Y_{p, k} = X_{p, k}$ for all k. [Set these components in any feasible manner] $(p \neq 1, 2)$

OUTPUT $Z = (Y - X)$ (Note: $U_{2, j} = 1$, $U_{2,k} = 0$ for all $k < j$)

end for

Repeat STEP 3 for each block $m = 3, 4, ..., F$

STEP 3:

For $j = 1, 2, ..., (T (1.2)$

Set vectors X and Y via:

if $1 \leq j \leq (T - w + 1)$

Block 1: $Y_{1, j} = 1$ $X_{1, j} = 1$ $Y_{1, k} = X_{1, k} = 0$, else.

Block 2: $Y_{2, (j+w(1.2)} = 1$ $X_{2, (j+w(1.2)} = 1$ $Y_{2, k} = X_{2, k} = 0$, else.

Block m: $Y_{m, j} = 1 X_{m, (j + 1)} = 1$ $Y_{m, k} = X_{m, k} = 0$, else.

else

Block 1: $Y_{1, j} = 1$ $X_{1, j} = 1$ $Y_{1, k} = X_{1, k} = 0$, for $k \neq j$

Block 2: $Y_{2, T} = 1$ $X_{2, T} = 1$ $Y_{2, k} = X_{2, k} = 0$ for $k \neq T$

Block m: $Y_{m, j} = 1$ $X_{m, (j+1)} = 1$ $Y_{m, k} = 0$ for $k \neq j$ and $X_{m, k} = 0$ for $k \neq j+1$

Block p: $Y_{p,k} = X_{p,k}$ for all k. (Set these components in any feasible manner)

$(p \neq 1, 2, m)$

Output row vector $U = (Y - X)$.

end for

Block p: $Y_{p,k} = X_{p,k}$ for all k. (Set these components in any feasible manner)

$(p \neq 1, 2, m)$

Output row vector $U = (Y - X)$.

end for

Proof that Algorithm 1 is correct:

We form a matrix, A, by letting the k^{th} row of matrix A be the k^{th} (row) vector output by Algorithm 1. Note that A has FT-many columns. To show that the algorithm is correct, it will suffice to show that the rows of A (output of the algorithm) are linearly independent, linear combinations of vectors from *S*, where *S* is the set of integer solutions in GHB_1^C. To this end, we will show three things:

(i) The number of rows in *A* is $n = (FT - F)$

(ii) The rows of *A* are linearly independent

(iii) Each row of *A* is a linear combination of vectors from *S*

Proof of (i) : Step 1 yields *T* vectors. Step 2 yields $(T - 2)$ vectors. Step 3 yields $(T - 1)$ vectors for each of its $(F - 2)$-many executions. The total number of vectors output by the algorithm is given by:

$$T + (T - 2) + (F - 2)(T - 1) = F(T - 1) = FT - F = n$$

Note: We have not shown that these vectors are distinct, but this will follow from the linear independence of the vectors.

Proof of (ii): To show that the rows of *A* are linearly independent, it will suffice to show that *A* is in row-echelon form and that each row is a pivot row. By construction, every (row) vector has a lead "1" in component (i, j), for some (i, j). We will show that the lead entries in the rows of matrix *A* are staggered, left to right. Let *U* be any vector output by the algorithm except the last. Suppose that the lead entry of *U* occurs in the position (i, j) (i.e., $U_{i\,j} = 1$ and for all $m < j$, $U_{i,m} = 0$ and for all $k < i$ and all h, $U_{k\,h} = 0$). The lead-entry of the next vector, U^*, will occur either in the same block i, and the position $(i, j+1)$, (whenever U^* is created in the same for-loop) or it will occur in the next block, $(i + 1)$ (whenever U^* is created in the subsequent for-loop). In either case, the lead-entry of U^* is strictly to the right of the lead entry in *U*. So, *A* is in row-echelon form and each of its n-many rows is a pivot row.

Proof of (iii): Lastly, we must show that each row of *A* is a linear combination of vectors in *S*. Each vector, *U*, output by the algorithm is formed by either $U = (Y - X)$ or $U = Y$, so, clearly, U is a linear combination of the vectors X and Y. Next, we must show that X and Y are in S. That is, we must show that X and Y are

(a) integer vectors

(b) solutions to *GHB* and

(c) meet constraint C at equality

Since the components of X and Y are binary, (a) is clear. Over each block, the components of Y (and X) sum to one, so constraints (1.3) are satisfied. Let j be $1 \leq j \leq T$. The number of k for which $Y_{k\,j} = 1$ (or $X_{k\,j} = 1$) is less than or equal to two (except for $j = T$) and since we have assumed that for all t, $b_t \geq 2$, Y (and X) satisfies the capacity constraints, (4). Let $Y_{i,\,j} = 1$ and $Y_{m,\,n} = 1$ where $m \neq i$. By construction, $|j - n| \leq w_b$, so Y satisfies the banking constraints. The same holds for X. So, X and Y are solutions to *GHB*, and (b) is shown. Finally, to show (c), note that for any vector, X, constraint C reads

$$\sum_{s=1}^{t} X_{1,s} + \sum_{s=t+w_b}^{T} X_{2,s} \leq 1.$$

To show that X meets C at equality, it suffices to show that exactly one of the following holds true:

(i) $X_{1,\,s} = 1$ for exactly one $s \in \{1, 2, ..., t\}$

(ii) $X_{2,\,s} = 1$ for exactly one $s \in \{t + w_b, t + w_b + 1, ..., T\}$

By this technique, we will show that, at each step of the algorithm, both X and Y meet C at equality.

Let $w = w_b$.

STEP 1:
for $j = 1, 2, ..., t$,

$Y_{1,\,j} = 1 \Rightarrow (1.2)$ true for Y

$Y_{2,\,j+w-1} = 1 \Rightarrow Y_{2,\,k} = 0$ for all $k \geq t + w \Rightarrow (1.3)$ false for Y

for $j = (t + 1), (t + 2), ..., T$

$Y_{1,j} = 1 \Rightarrow Y_{i,k} = 0$ for all $k \leq t \Rightarrow (1.2)$ false for Y

$Y_{2,T} = 1 \Rightarrow (1.3)$ true for Y

STEP 2:

for j $= 1, 2, ..., w-2 + t$

$Y_{1,j} = 1$ for some $1 \leq j \leq t \Rightarrow (1.2)$ true for Y

$Y_{2,j} = 1 \Rightarrow Y_{2,k} = 0$ for all $k \geq t + w \Rightarrow (1.3)$ false for Y

$X_{1,j} = 1$ for some $1 \leq j \leq t \Rightarrow (1.2)$ true for X

$X_{2,j+1} = 1 \Rightarrow X_{2,k} = 0$ for all $k \geq t + w \Rightarrow (1.3)$ false for X

for $j = (w + t), (w + t) + 1, ..., (T - 1)$

$Y_{1,j} = 1 \Rightarrow (1.2)$ false for Y

$Y_{2,u} = 1$ for $u \geq w + t \Rightarrow (1.3)$ true for Y

$X_{1,j} = 1 \Rightarrow (1.2)$ false for X

$X_{2,u} = 1$ for $u \geq w + t \Rightarrow (1.3)$ true for X

STEP 3:

Note that $Y_{1,k} = X_{1,k}$ and $Y_{2,k} = X_{2,k}$ for all k

for $j = 1, 2, ... t$

$Y_{1,j} = 1 \Rightarrow (1.2)$ true for Y, (1.2) true for X

$Y_{2,j+w-1} = 1 \Rightarrow Y_{2,k} = 0$ for all $k \geq t + w \Rightarrow (1.3)$ false for Y, (1.3) false for Y

for $j = t, t + 1, ..., T$

$Y_{1,j} = 1 \Rightarrow (1.2)$ false for Y, (1.2) false for X

$Y_{2,u} = 1$ for some $u \geq w + t \Rightarrow (1.3)$ true for Y, (1.3) true for X

Thus, each X and each Y produced by algorithm 1 satisfy C at equality. In all, we have shown that there are (at least) n-many linearly independent (hence, affinely independent), integer vectors in GHB_1^C that meet constraint C at equality.

•

Proof of Lemma 2: Recall that in Lemma 1, we generated a matrix A of n-many linearly independent, integer vectors from span(Ω), where $\Omega \subseteq GHB_1^C$. We will show how to add to matrix A one more linearly independent, integer vector from GHB_1^C for a total of (n + 1)-many linearly independent integer solutions to GHB_1^C. Since linearly independent vectors are affinely independent, this will show that dim(GHB_1^C) ≥ n.

As it stands, matrix A does not have a row with a pivot in component (2, k) (i.e., the k^{th} component of the second block), where $k = (w_b + t$ (1.2). But we can generate such a row by creating a vector, $U = (Y - X)$, where Y and X are integer solutions to SAGHPBC and constructed as follows:

In block 1: $Y_{1,k} = X_{1,k} = 1$ all other components are zero

In block 2: $Y_{2,k} = 1$ $X_{2,k+1} = 1$ all other components are zero

In block m ($m \neq 1, 2$): $Y_{m,n} = X_{m,n}$ for all n. Set these binary components in any feasible manner.

Note that, since $U = (Y - X)$, $U_{2,k} = 1$ and all components to the left of $U_{2,k}$ are zero. Because of its unique pivot, this row is linearly independent of the other rows.

•

Proof of Lemma 3:

Let $w = w_b$.

Case 1: (4.2) with $\tau > t$. Since $Y_{t+w} = 1$, $Y_s = 0$ for all $s > t + w$. Thus, $\tau + w > t + w$ implies that $\sum_{i=\tau+w}^{T} \mathrm{Y}_{\mathrm{i}} = 0$.

Case 2: (4.2) with $\tau < t$. Since $X_t = 1$, $X_s = 0$ for all $s < t$ and we have that $\sum_{i=1}^{\tau} X_i = 0$.

Case 3: (4.3) with $\tau < t + w$. Since $Y_{t+w} = 1$, $Y_s = 0$ for all $s < (t + w)$, and we have that $\sum_{i=1}^{\tau} Y_i = 0$.

Case 4: (4.3) with $\tau \geq t + w$. Since $X_t = 1$, we see that $X_s = 0$ for all $s > t$. In particular, $X_s = 0$ for all $s \geq \tau$.

•

Proof of Theorem 4:

Let C be an arbitrary constraint of the form (2.9). Then for some time interval t, and some flight f, C has the form

$$\sum_{i=t}^{T} Z_i^b - \sum_{i=t}^{T} X_{fi} \leq 0. \tag{B.1}$$

None of our work is affected by the assignment of a flight outside bank b to the last time interval, T. Thus, for ease of vector notation, we can ignore all flights not in bank b and assume that the set of flights, $\{1, 2, \ldots, F + B\}$is indexed so that variable Z corresponds to flight 1 (i.e., flight 1 is the "ghost flight") and that variable X_{ft} corresponds to flight 2.

The proof is almost identical to the proof of Theorem 1. All of the vectors constructed in Algorithm 1 (with $n = n^*$) are in GHB_2^C. All but five of those vectors meet constraint (B.1) at equality. Below are the replacements necessary so that all vectors X and Y generated in (but not output by) algorithm 1 meet (B.1) at equality.

In STEP 1, iteration $j = (t - 1)$, change vector Y so that

$Y_{1, t\_1} = Y_{2, t\_1} = 1.$

In STEP 2, iteration $j = (t - 1)$, change vectors Y and X so that

$Y_{1, t+1} = Y_{2, t+1} = X_{1,t+1} = X_{2, t+2} = 1.$

In STEP 3, iteration $j = (t - 1)$, change vectors Y and X so that

$Y_{1, t\_1} = Y_{2, t\_1} = X_{1, t\_1} = X_{2, t\_1} = Y_{3, t\_1} = X_{3, t} = 1.$ w_b

With these minor modifications to algorithm 1, all of its output is in the span of the set of vectors that meet (B.1) at equality. Thus, there are at least n^*-many linearly independent (affinely independent) vectors that meet (B.1) at equality and the face, F, represented by (B.1) must have dimension at least ($n^* - 1$). (B.1) is the one and only constraint to eliminate the solution, X, in which flight f lands in time slot ($t - 1$) and the ghost flight lands in time slot t. Therefore, dim(F) = ($n^* - 1$), and since dim(GHB_2^C) = n^* (see Prop 3), F is a facet of GHB_2^C. Moreover, the uniqueness of X implies that (B.1) is the only constraint of its kind that represents F.

•

Proof of Theorem 6: Fix a time interval $t \neq T$ and let C be the corresponding capacity constraint. Let $k = MAX_{t \neq T}(b_t)$. By re-indexing or adding dummy flights to the set of flights, F, we may assume that the last (k +1)-many flights of F is a set, F^*, of non-bank flights such that a_f = 1, for each $f \in F^*$. Let F^* be indexed via $\{f_1, f_2, ..., f_k, f_{k+1}\}$.

Recall that the components of each N^*-dimensional vector are indexed by the set

$$I = \{ (i, j) : 1 \le i \le F \text{ and } 1 \le j \le T \}.$$

We define $I^* \subseteq I$ via $I^* = \{ (i, j) \in I : j \neq T\}$. Note that there is one pair in I^* for every component of every flight block except the last component. Thus, $|I^*| = N^* - F = n^*$. For each $(i, j) \in I^*$, we will generate a row vector U with a lead "1" (all zeros to the left) in component (i, j). The set Ω will be the collection of all such U-vectors. Thus, $|\Omega| = n^*$ and the vectors in Ω are linearly independent because they can be used to form the rows of an upper triangular matrix.

Fix the index (i, j).

Case 1: $i \in (F - F^*)$. First, generate a feasible solution vector, Y as follows. Assign flight i to time interval j. If i is a bank flight in, say, bank b, then let w_b be the width of the bank b. Assign the flights of $(\Phi_b - \{i\})$ to time intervals j , $j+1$, ..., $j+ w_b$ in any feasible manner that does not exhaust the capacity of interval (j+1) (this uses Assumption 2). If $j = t$, then assign the last $(b_t - 1)$ flights of F^* to time interval t. If $j \neq t$, then assign the last b_t flights of F^* to time interval t. Assign every flight in $(F - \Phi_b)$, including the remaining flights of F^*, to time interval T. Note that whether $j = t$ or $j \neq t$, there are exactly b_t flights assigned to time interval t and that there is exactly one flight assigned to time interval j. Vector Y meet the capacity constraint at equality, hence, is in F_t.

Secondly, generate a feasible solution (row) vector X by setting every component of X as in Y, except that flight i should be assigned to time interval (t +1) (this is possible by assumption 2). If $j = t$, then one of the flights that is currently assigned to time interval

t should be reassigned to time interval T. Thus, there will be exactly b_t-many flights assigned to interval t. Vector X meets the capacity constraint at equality, hence, is in F_t.

Let $U = (Y - X)$. Clearly, U is a linear combination of vectors in F_t. Y and X are the same in all components strictly to the left of (i, j). Moreover, $Y_{i,j} = 1$, $X_{i,j} = 0$, $Y_{i,j+1} = 0$, $X_{i,j+1} = 1$. Thus, $U_{i,j} = 1$, $U_{i,j+1} = -1$ and all other components of U are zero. U has a lead "1" in component (i, j), as desired.

Case 2: $i \in F^* = \{f_1, f_2, ..., f_k, f_{k+1}\}$.

If $i = f_1$, then construct row vectors Y and X as follows.

Vector Y: Assign all flights of $(F - F^*)$ to time interval T. Assign f_1 to time interval j. If $j \neq t$, then assign b_t-many of the flights of $(F^* - \{f_1\})$ to time interval t. This is possible because $|F^*| = (k+1)$, where $k = b_t$. And if $j = t$, assign $b_t - 1$ flights of F^* to time interval t. Either way, the number of flights assigned to time interval t is b_t and the vector Y is in F_t.

Vector X: Assign f_1 to time interval $(j+1)$. If $(j+1) \neq$ t, then assign b_t-many of the flights of $(F^* - \{f1\})$ to time interval t. And if $j = t$, then assign $b_t - 1$ flights of F^* to time interval t. Either way, the number of flights assigned to time interval t is b_t and the vector X is in F_t.

If $i > f_1$, then construct row vectors Y and X as follows.

Vector Y: Assign all flights of $(F - F^*)$ to time interval T. Assign flight i to time interval j. If $j \neq t$, then assign b_t-many of the flights of $(F^* - \{i\})$ to time interval t. If $j = t$, then let $d = b_t - 1$ and assign d-many of the flights to time interval t. Assign all remaining flights of $(F^* - \{i\})$ to time interval T.

Vector X: For each $i < F_1$ and for each j,

Let $U = (Y - X)$. Clearly, U is a linear combination of vectors in F_t. Y and X are the same in all components strictly to the left of (i, j). Moreover, $Y_{i,j} = 1$, $X_{i,j} = 0$, $Y_{i,j+1} = 0$, $X_{i,j+1} = 1$. Thus, $U_{i,j} = 1$, $U_{i,j+1} = -1$ and all other components of U are zero. U has a lead "1" in component (i, j), as desired.

TABLE 1: Data Set 1									
Flights = 25		Banks = 2 (6 each) 48%			Width = 3	Capacity = 5		T = 6	
		LP			IP				
Model	Gap (%)	Function value	Iterations Simplex	Time (sec)	Function value	Iterations Simplex	Time (sec)	Nodes B&B	
XTC	3.41	90.60	87	0.13	93.80	382	0.70	93	
XW	1.81	92.10	113	0.13	93.80	209	0.30	18	
XWZ	1.81	92.10	113	0.12	93.80	345	0.53	4	
XMM	11.64	82.88	111	0.17	93.80	1991	4.75	776	
XMMZ	11.64	82.88	111	0.18	93.80	231	0.30	6	
XSS	0.00	INT 93.80	92	0.15	93.80	92	0.15	0	
WSS	0.00	INT 93.80	172	0.23	93.80	172	0.23	0	
XGF	0.00	INT 93.80	93	0.12	93.80	93	0.15	0	
WGF	0.00	INT 93.80	158	0.18	93.80	158	0.18	0	

TABLE 2: Data Set 2									
Flights = 25		Banks = 2 (9 each) 79%			Width = 3	Capacity = 5		T = 6	
		LP			IP				
Model	Gap (%)	Function value	Iterations Simplex	Time (sec)	Function value	Iterations Simplex	Time (sec)	Nodes B&B	
XTC	11.35	100.35	124	0.27	113.20	3773	0.27	629	
XW	8.82	103.22	159	0.28	113.20	1352	0.27	156	
XWZ	8.82	103.22	159	0.28	113.20	268	0.27	8	
XMM	25.22	84.65	118	0.20	113.20	23904	64.10	6161	
XMMZ	25.22	84.65	118	0.20	113.20	226	0.42	4	
XSS	2.32	110.57	112	0.30	113.20	139	0.42	8	
WSS	2.32	110.57	224	0.53	113.20	290	0.82	26	
XGF	2.32	110.57	115	0.18	113.20	125	0.27	3	
WGF	2.32	110.57	176	0.30	113.20	197	0.35	2	

TABLE 3: Data Set 3									
Flights = 79		Banks = 2 (10 each) 25%			Width = 3	Capacity = 5		T = 16	
		LP			IP				
Model	Gap (%)	Function value	Iterations Simplex	Time (sec)	Function value	Iterations Simplex	Time (sec)	Nodes B&B	
XTC	0.47	890.92	585	1.30	895.10	2932	20.22	540	
XW	2.96	868.62	961	3.42	NL 934.40	388,988	2021.37	20,000	
XWZ	2.96	868.62	967	3.93	895.10	1561	6.27	16	
XMM	5.37	847.00	596	3.33	NL 920.10	134,303	924.90	20,000	
XMMZ	5.37	847.00	596	3.35	895.10	1548	7.38	14	
XSS	0.31	892.35	720	8.13	895.10	4645	86.47	574	
WSS	0.31	892.35	1937	17.60	895.10	4343	50.88	382	
XGF	0.31	892.35	657	3.30	895.10	694	3.40	3	
WGF	0.31	892.35	1683	7.68	895.10	1733	10.05	5	

INT - integer solution in LP LIMIT - 3 hour CPU time limit reached time
N/A - not applicable (limits reached) NL - node limit reached (20,000)

TABLE 4: Data Set 4								
Flights = 120		Banks = 4 (8 each) 26%		Width = 3	Capacity = 5		T = 24	
		LP			IP			
Model	Gap (%)	Function value	Iterations Simplex	Time (sec)	Function value	Iterations Simplex	Time (sec)	Nodes B&B
XTC	0.69	2784.71	1369	4.73	NL 2915.20	437,068	4171	20,000
XW	4.90	2666.72	2525	16.08	NL 2898.20	726,325	8556	20,000
XWZ	4.90	2666.72	2525	16.08	2804.10	17502	191	283
XMM	7.22	2601.53	1637	21.18	NL 3001.60	279,618	3801	20,000
XMMZ	7.22	2601.53	1637	21.18	2804.10	11980	149	146
XSS	0.26	2796.68	1838	38.95	NL 2804.60	398,102	11,589	20,000
WSS	0.26	2796.68	7060	142.38	NL 2804.80	464,154	11,958	20,000
XGF	0.26	2796.68	1845	14.22	2804.10	2924	30	24
WGF	0.26	2796.68	4489	26.55	NL 2804.80	5478	64	26

TABLE 5: Data Set 5A (13:00-16:59)								
Flights = 280		Banks = 6 (12-36) 48%		Width = 6	Capacity = 10		T = 30	
		LP			IP			
Model	Gap (%)	Function value	Iterations Simplex	Time (sec)	Function value	Iterations Simplex	Time (sec)	Nodes B&B
XTC	0.00	7317.78	1996	29.17	7318.10	2002	30.43	3
XW	0.09	7311.60	6784	103.42	7318.10	8938	156.33	140
XWZ	0.09	7311.60	6784	103.90	7318.10	6805	104.02	4
XMM	0.41	7287.81	2110	55.35	7318.10	61,411	3807.30	20,000
XMMZ	0.41	7287.81	2110	55.63	7318.10	2225	63.48	3
XSS	0.00	INT 7318.10	6925	1132.73	7318.10	6925	1127.28	0
WSS	0.00	INT 7318.10	47,250	9037.80	7318.10	47,250	9024.78	0
XGF	0.00	INT 7318.10	3875	71.82	7318.10	3875	71.63	0
WGF	0.00	INT 7318.10	12,708	292.30	7318.10	12,708	292.30	0

TABLE 6: Data Set 5B (13:00-18:59)								
Flights = 419		Banks = 10 (12-36) 48%		Width = 6	Capacity = 10		T = 42	
		LP			IP			
Model	Gap (%)	Function value	Iterations Simplex	Time (sec)	Function value	Iteration Simplex	Time (sec)	Nodes B&B
XTC	0.05	14,579.28	3533	39.63	14,587.10	5276	150.82	323
XW	0.74	14,478.50	23,325	381.70	NL 14,620.40	326,130	7720.70	20,000
XWZ	0.74	14,478.50	23,325	381.10	14,587.10	23,892	383.90	16
XMM	1.67	14,343.19	5224	137.83	NL 14,647.60	158,201	10,559.28	20,000
XMMZ	1.67	14,343.19	5224	138.15	14,587.10	6302	286.23	29
XSS	N/A	N/A	N/A	LIMIT	N/A	N/A	LIMIT	N/A
WSS	< 0.01	INT 1458.71	34,701	2137.72	14,587.10	34,701	2136.73	0
XGF	< 0.01	INT 1458.71	22,052	1590.52	14,587.10	22,052	1508.53	0
WGF	< 0.01	INT 1458.71	34,701	2137.72	14,587.10	34,701	2138.42	0

INT - integer solution in LP LIMIT - 3 hour CPU time limit reached time
N/A - not applicable (limits reached) NL - node limit reached (20,000)

TABLE 7: Data Set 5C (13:00-20:59)								
Flights = 536		Banks = 6 (12-36) 45%		Width = 6	Capacity = 10		T = 54	
		LP			IP			
Model	Gap (%)	Function value	Iterations Simplex	Time (sec)	Function value	Iterations Simplex	Time (sec)	Nodes B&B
XTC	0.06	22,822.05	6504	187.93	22,835.90	11,725	1360.90	1368
XW	0.82	22,647.80	49,186	1888.05	N/A	N/A	LIMIT	N/A
XWZ	0.82	22,647.80	49,186	1886.98	22,835.90	50,620	2360.15	35
XMM	1.72	22,442.19	7973	531.82	N/A	N/A	LIMIT	N/A
XMMZ	1.72	22,442.19	7973	533.02	22,835.90	11,460	892.77	48
XSS	N/A	N/A	N/A	LIMIT	N/A	N/A	LIMIT	N/A
WSS	N/A	N/A	N/A	LIMIT	N/A	N/A	LIMIT	N/A
XGF	0.03	22,829.87	14,944	1197.22	22,835.90	15,030	1205.31	3
WGF	N/A	22,829.87	81,633	8266.10	22,835.90	81,718	8296.67	2

INT - integer solution in LP
LIMIT - 3 hour CPU time limit reached time
N/A - not applicable (limits reached)
NL - node limit reached (20,000)

TABLE 8 Does LP performance improve with a uniform bound on flight tardiness?						
Data Set	Model	Time Intervals	Uppe r bound	Cap	Iterations Simplex	Time (sec)
4 (before bound)	XGF	24	none	7	923	5.63
4 (after bound)	XGF	24	6	7	671	2.17
			Improvement:		27.30%	61.46%
4 (before bound)	WGF	24	none	7	3919	42.00
4 (after bound)	WGF	24	6	7	3067	16.32
			Improvement:		21.74%	61.11%
5A (before bound)	XGF	30	none	10	3875	80.25
5A (after bound)	XGF	30	5	10	1450	9.20
			Improvement:		62.58%	88.88%
5A (before bound)	WGF	30	none	10	12,708	292.30
5A (after bound)	WGF	30	5	10	7061	83.75

References

[1] ANDREATTA, G., and G. ROMANIN-JACUR, 1987. "Aircraft Flow Management Under Congestion". *Transportation Science*. **21**, 249-253.

[2] ANDREATTA, G., ODONI, A.R. and RICHETTA, O. 1993, "Models for the Ground-Holding Problem", in *Large-Scale Computation and Information Processing in Air Traffic Control*, L. Bianco and A.R. Odoni, eds., Springer-Verlag, Berlin, pp. 125-168.

[3] ATTWOOL, V.W. 1977. "Some Mathematical Aspects of Air Traffic Systems". *J. Inst. Navig*. **30**, 394-411

[4] BALL, M. O., LIU, W. and PULLEYBLANK W.R.. 1989. "Two-Terminal Steiner Tree Polyhedra", in *Contributions to Operations Research and Economics: the Twentieth Anniversary of CORE*, MIT Press, Cambridge, Mass, 251-284.

[5] BERTSIMAS, D. and STOCK, S. 1996. "Air Traffic Flow Management Problem with Enroute Capacities", to appear in *Operations Research*.

[6] NEMHAUSER, G.L. and WOLSEY, L., 1988, *Integer and Combinatorial Optimization*, Wiley, New York.

[7] ODONI, A. R.. 1987. "The Flow Management Problem in Air Traffic Control," in *Flow Control of Congested Networks*, 269-288, A. R. Odoni, L. Bianco and G. Szego (eds.), Springer Verlag, New York.

[8] PULLEYBLANK, W. R.. 1989. "Polyhedral Combinatorics", in *Handbook on Operations Research and Management Science, Volume 1: Optimization*, G.L. Nemhauser, A.H.G. Rinnooy Kan and M.J. Todd, eds., North-Holland, Amsterdam.

[9] RICHETTA, O. and A.R. ODONI. 1993. "Solving Optimally the Static Ground-Holding Policy Problem in Air Traffic Control," *Transportation Science* **27**, 228-238.

[10] RICHETTA, O. 1995. "Optimal Algorithms and a Remarkably Efficient Heuristic for the Ground-Holding Problem in Air Traffic Control," *Operations Research* **43**, 758-770.

[11] SOKKAPIA, B. G. 1985. *Arrival Flow Management as a Feedback Control System.* The Mitre Corporation, Washington, D.C.

[12] TERRAB, M. and A. R. ODONI. 1993. “Strategic Flow Management for Air Traffic Control,” *Operations Research* **41**, 138-152.

[13] VRANAS, P., D. BERTSIMAS and A. R. ODONI. 1994.“The Multi-Airport Ground-Holding Problem in Air Traffic Control,” *Operations Research* **42**, 249-261.

[14] VRANAS, P., D. BERTSIMAS and A. R. ODONI. 1994. “Dynamic Ground-Holding Policies for a Network of Airports”, *Transportation Science* **28**, 275-291.

[15] WANG, H. 1991. “A Dynamic Programming Framework for the Global Flow Control Problem in Air Traffic Management,” *Transportation Science* **25**, 308-313.