Optimization and Mediated Bartering Models for Ground Delay Programs

Michael Ball
Robert H. Smith School of Business
and Institute for Systems Research
University of Maryland
College Park, MD 20742 USA
Robert Hoffman
Metron Aviation, Inc.
131 Elden St.
Herndon VA 20170
hoffman@metsci.com
Thomas Vossen
Leeds School of Business
University of Colorado
Boulder, CO 80309

June 15, 2002

Abstract

1 Introduction

In this paper, we define and analyze several resource allocation methods based on the principles of Collaborative Decision Making (CDM). CDM procedures are currently employed within the US for the planning and control of ground delay programs (GDPs). Two resource allocation procedures, ration-by-schedule (RBS) and compression, are embedded with the CDM decision support tool, FSM. Since these procedures have been accepted by the key players involved and have undergone extensive scrutiny through "war-gaming" exercises and also several years of actual use, it is important to extract their essential features in planning future systems. At the same time, they should be critically analyzed with an eye toward improvement and extension to other settings. In Section 2 of the paper, we describe RBS and compression and summarize some of their important properties. In Sections 3 and 4, we examine two directions for enhancements. First, in Section 3, we argue that the allocation principles underlying RBS are "flight-centric" and that an "airline-centric" approach may be more appropriate. There are a number of ways in which the airline-centric philosophy may be applied. We examine one that addresses inequities across airline allocations that can arise due to the manner in which flights are exempted from ground delay programs based on the distance of the departure airport from the arrival airport where the GDP is based. We also examine the problem of resource allocation in the enroute airspace. In Section 4, we define some possible enroute extensions of RBS and provide results from a set of experiments that compare these methods. In particular, we investigate a natural extensive of RBS to the enroute setting, namely allocation based on accrued delay.

2 GDP Procedures

In the quest for efficient allocation of resources, it is tempting to turn to the scientific literature for one of the many mathematical models that determine the most efficient TFM initiative [cite some refs such as Bertsimas?]. The enhanced ground delay program (GDP-E) experience has shown that efficiency can be achieved only through equitable treatment of FAA customers on a macro level. Prior to CDM, effective GDP initiatives were based on dated flight data, suffused with omissions, redundancies, and outright errors. Exclusion of the air carriers from the decision process led to a breakdown of communication and cooperation between the FAA and the air carriers in ground delay programs. Once CDM provided incentives for air carriers to provide up-to-date flight data, carriers participate more in the TFM decision-making process, trust grows, and enhanced understanding leads to an infusion of improved data and user intent information. In turn, this allows for more effective and efficient decision making. By charter, the FAA is obligated to treat flights on an equal basis. One could argue that the primary concern of TFM should be the minimization of adverse impacts associated with demand-capacity imbalances on the passengers, rather than on individual aircraft. But even if TFM had access to accurate passenger count information, they would need extensive connectivity information to go along with it: equipment connections, flight crew schedules, and so on. Air carrier operations are not just collections of flight instances; they are carefully choreographed networks of circulating equipment and personnel. Disruption of services can easily spread across the NAS, and even into the next day. The best way to service the passengers is to allow the air carriers to check the growth of delays and to return to regular operational conditions as quickly as possible. The CDM-enhanced GDP procedures are the best example of this in practice: TFM distributes arrival slots to the carriers, who then mitigate ground delay damages by canceling flights, merging passengers, and swapping arrival slots in favor of those flights most critical to the health of their operation. This involves an intimate understanding of how each air carrier schedule is structured. These types of equity decisions and tradeoffs are best left to the carriers, for only they have the operational knowledge and expertise to mitigate damages. Once the airlines have reacted to TFM initiatives, the FAA specialists can fine-tune the TFM initiative by revising and/or compressing the program to improve results. CDM makes this proactive/reactive process possible.

Under CDM the interaction between the FAA and the airlines may be represented as shown in Figure 1. Initially, the FAA rations the arrival slots among the airlines. Given their slots, airlines may subsequently substitute and cancel flights. Next, the FAA executes Compression. In the remainder of this section we describe the RBS and Compression algorithms in greater detail. In doing so, we use the following notation:

- $\mathcal{F}$, the set of flights in the GDP. The mapping $O : \mathcal{F} \to \mathcal{A}$ defines the flight-airline relation. For each airline $a \in \mathcal{A}$, we use $\mathcal{F}_a$ represents the flights from that airline, i.e., $\mathcal{F}_a = \{f \in \mathcal{F} \mid O(f) = a\}$.
- $\mathcal{S} = \{1, \ldots, n\}$, the sequence of arrival slots in the GDP.

2.1 Ration-by-Schedule

As a first step in a GDP, the *Ration-By-Schedule* algorithm rations the arrival slots among airlines. As in Grover Jack, the RBS algorithm assigns flights to slots on a first-come, first-served basis. However, in RBS flights are ordered according to their *original scheduled time of arrival* (as opposed to the *most recent estimated time of arrival* ordering used in Grover Jack). As a consequence, airlines will not forfeit a slot by reporting a delay or a cancellation, which was the case when using the Grover Jack algorithm. Conceptually, the RBS algorithm can be outlined as follows.

RBS Algorithm

Step 1. Order the flights in $\mathcal{F}$ by their original scheduled time of arrival. Goto step 2.

Step 2. Select the first flight in $\mathcal{F}$ that has not been assigned a slot. If there is none, the algorithm is terminated. Otherwise, the flight is assigned to the first unassigned slot it can meet (according to its original scheduled time of arrival).

The actual RBS algorithm has to take into account several complicating factors, such as flights being airborne, flights exempted from the GDP, and the possibility that a

Figure 1: FAA/airline interaction in a GDP under CDM

GDP was already executed before. (see Hoffman et al., 2000 for a discussion of these details).

It should be noted that the resulting flight schedule may be inefficient in its utilization of arrival capacity. Arrival slots may have been assigned to canceled flights or to flights that are delayed and cannot use the slot. However, the end result of RBS should not be viewed as an assignment of slots to flights but rather as an assignment of slots to airlines. After RBS through the cancellation and substitution process, an airline can reassign the slots it owns to any of its flights. It should be emphasized that this notion of slot ownership is one of the main tenets of the CDM paradigm, and there is a general consensus among airlines that this is indeed a fair method of rationing arrival capacity.

2.2 Compression

After a round of substitutions and cancellations the utilization of slots can usually be improved. The reason for this is that an airline's flight cancellations and delays may create "holes" in the current schedule; that is, there will be arrival slots which have no flights assigned to them. The purpose of the Compression algorithm is to move flights up in the schedule to fill these slots. The basic idea behind the compression algorithm is that airlines are "paid back" for the slots they release, so as to encourage airlines to report cancellations. The extent to which a flight can be moved up will be limited, e.g. a flight cannot depart before its scheduled departure time. To capture this, each flight has an *earliest arrival time* specified by the mapping

- $e : \mathcal{F} \to \mathcal{S}$.

Moreover, it is assumed that a flight cannot be moved down from its position in the current schedule I. Thus, the set of slots $\{e(f), \ldots, I(f)\}$ defines the window in which flight f can land. Now, the compression algorithm can be outlined as follows.

Compression Algorithm

Step 1. Determine the set of open slots $\mathcal{C}_\mathcal{S}$. For each slot $c \in \mathcal{C}_\mathcal{S}$, execute step 2.

Step 2. Determine the owner of slot c, that is, the airline a that owns the canceled or delayed flight f that has been assigned to slot c. Try to fill slot c, according to the following rules:

1. Determine the first flight g from airline a (in the current schedule) that can be assigned to slot c, that is, for which $c \in \{e(g), \ldots, I(g)\}$. If there is no such flight, go to step 2.2. Otherwise, execute Step 3.
2. Determine the first flight g from any other airline that can be assigned to slot c. If there is no such flight, goto step 2.3. Otherwise, execute Step 3.
3. There is no flight that can be assigned to slot c. Therefore, return to Step 1 and select the next open slot.

Step 3. Swap the slot assignments of flights f and g, i.e, assign flight g to slot i, and flight f to slot $I(g)$ (the slot occupied by by g). Note that airline a is now the owner of (open) slot $I(g)$. Next, slot $I(g)$ is made the current slot, and Step 2 is repeated.

The algorithm terminates when Step 2 has been executed for all open slots in $\mathcal{C}_{\mathcal{S}}$.

The important features of the compression algorithms are that (i) arrival slots are filled whenever possible, (ii) flights from the airline that owns the current open slot c are considered before all others, (iii) if the controlling airline cannot use a slot it is compensated by receiving control over the slot vacated by the flight which moves into its slot, and (iv) airlines do not involuntarily lose slots they own and can use.

2.3 Priority Based on Accrued Delay

Recently, (see Howard, 2000) a new criterion was proposed for the fair allocation of air traffic resources. This criterion was specifically aimed at allocation within the enroute airspace. In this section, we formalize this criterion for the case of allocating GDP arrival time slots. We will show in Section 3.3 that this criterion is equivalent to RBS for the case of allocating arrival time slots within a GDP.

First, we state the (informal) statement of the criterion:

> if two or more flights are competing for the same resource, then allocate the resource to that flight with the largest accrued delay.

Here "accrued delay" is not precisely defined, but it is meant to include delay from all possible sources, e.g. a mechanical delay experienced by the aircraft, air traffic control imposed delay, etc. It would be reasonable to calculate accrued delay as follows:

> (expected destination arrival time of the flight from the time at which the resource is to be allocated assuming no additional delays) - (scheduled destination arrival time)

We now formally state a GDP allocation procedure which embodies these ideas. Suppose again that $\mathcal{F} = \{f_1, \ldots, f_n\}$ represents the flights in the GDP, and $\mathcal{S} = \{s_1, \ldots, s_n\}$ the available slots. For each flight f , we define oag_f to be its originally schedules arrival time, eta_f to be its (current) estimated time of arrival, and cta_f to be its controlled time of arrival. With each slot s we associate a time t_s. The algorithm statement is given below.

Accrued-Delay-Based Slot Allocation Procedure

Step 0. Let $\hat{\mathcal{S}} := \mathcal{S}$ represent the unassigned slots, and $\hat{\mathcal{F}} := \mathcal{F}$ represent the unassigned flights.

Step 1. Select the first unassigned slot, that is, let $s := \arg\min_{s \in \hat{\mathcal{S}}} t_s$. If all slots have been assigned, terminate. Otherwise, let $\hat{\mathcal{S}} := \hat{\mathcal{S}}/\{s\}$ and go to step 2.

Step 2. Among all flights that can be assigned slot s, select the flight f with the largest amount of delay, that is, let

$$f := \arg \max_{f \in \hat{\mathcal{F}} : eta_f \leq t_s} (t_s - oag_f).$$

Let $cta_f := t_s$ (i.e. assign flight f to slot s), and let $\hat{\mathcal{F}} := \hat{\mathcal{F}}/\{f\}$. Go to step 1.

3 Optimization Models for Slot Allocation and Slot Exchange

In the slot allocation and exchange procedures used in CDM, the notion of fairness is largely implicit in the RBS and Compression algorithms. In this section we aim to make these fairness concepts explicit by formulating slot allocation and exchange as optimization models. Here, fairness is achieved by an appropriately defined objective derived from goal programming techniques. We compare the optimization models with RBS and Compression, and discuss possible advantages of using an optimization approach.

3.1 OPTIFLOW model

The OPTIFLOW model (see Ball, 1993) is an optimization model that assigns flights to slots in such a way that overall delay costs are minimized. As such, the model can be said to be representative of approaches that follow a central planning paradigm, and therefore not directly applicable to a decentralized approach. Still, the optiflow model can be considered the basis for all the optimization models described later in this section.

In order to formulate the OPTIFLOW model, we let $\mathcal{F}$ be the set of flights in the GDP, $\mathcal{S}$ the set of arrival slots, and $\mathcal{C}_\mathcal{F}$ the set of canceled flights in the algorithm, as defined in the previous section. The OPTIFLOW model is formulated as an assignment problem, with

- decision variables x_{ij}, for all $i \in \mathcal{F}/\mathcal{C}_\mathcal{F}, j \in \{e(i), \ldots, I(i)\}$. $x_{ij} = 1$ if flight i is assigned to slot j, and $x_{ij} = 0$ otherwise.
- cost coefficients c_{ij}, for all $i \in \mathcal{F}/\mathcal{C}_\mathcal{F}, j \in \{e(i), \ldots, I(i)\}$. c_{ij} represents the "cost" of assigning flight i to slot j.

The Linear Programming formulation of the problem is as follows.

$$\begin{array}{lll} \text{Min} & \sum_{i \in \mathcal{F}/\mathcal{C}_\mathcal{F}, j \in \{e(i), \ldots, I(i)\}} c_{i,j} x_{ij} & \\ \text{subject to:} & & \\ & \sum_{j \in \{e(i), I(i)\}} x_{ij} = 1 & \text{for all } i \in \mathcal{F}/\mathcal{C}_\mathcal{F} \\ & \sum_{i \in \mathcal{F}/\mathcal{C}_\mathcal{F}} x_{ij} \leq 1 & \text{for all } i \in \mathcal{S} \\ & x_{i,j} \in \{0, 1\} & i \in \mathcal{F}/\mathcal{C}_\mathcal{F}, j \in \{e(i), I(i)\} \end{array}$$

The constraints express that that each flight that is not canceled is assigned to a slot, and that each slot is assigned to at most one flight.

In the OPTIFLOW model, cost coefficients are expressed as $c_{ij} = w_i(j - I(j))^{1+\epsilon}$, with w_i a weight associated with flight i and $0 < \epsilon < 1$. Parameter ϵ is used for super-linear growth in the cost of the tardiness of a flight, so that the model tends to favor assigning a moderate amount of delay to two flights rather than assigning a large amount of delay to one flight and a small amount to another. For example, suppose that two flights are assigned 120 minutes of delay in total. If the choice is between assigning one flight a 30 minute delay and the other a 90 minute delay, or assigning both flights 60 minutes of delay, the model will choose the latter.

Traditionally, the weights w_f in the cost coefficients represent the cost associated the ground and airborne delay for that flight. Naturally, the flight schedules obtained by the model will depend heavily on these weights. However, a basic characterization of the flight schedules achieved by the OPTIFLOW model under various weights w_f is stated in the following proposition.

Theorem 3.1. *If $w_f > 0$ for all $f \in \mathcal{F}$, the OPTIFLOW model will find a solution that minimizes overall delay. Moreover, the slots that have flights assigned to them are identical for any delay-minimizing solution.*

Proof. First, observe that any solution that minimizes overall delay is an optimal solution to the assignment problem P' with cost coefficients $c'_{ij} = j - e_i$.

Now, let x^* be an optimal solution to the OPTIFLOW model, but suppose that it not optimal to P'. Thus, x^* does not minimize overall delay.

To show that this results in a contradiction, let y^* be an optimal delay minimizing solution (to P') and consider the symmetric difference $x^* \oplus y^*$ of the arcs in x^* and y^*. It is easy to see that the graph induced by $x^* \oplus y^*$ consists of a collection of even alternating cycles and even alternating paths. Since any alternating cycle represents assignments of the same set of flights to the same set of slots, the total delay of its flights is the same under both x^* and y^*. More specifically, for any cycle C we have

$$\sum_{(i,j) \in x^* \cap C} c'_{ij} = \sum_{j \in \mathcal{S} \cap C} j - \sum_{i \in \mathcal{F} \cap C} e_i = \sum_{(i,j) \in y^* \cap C} c'_{ij}.$$

Thus, there must be at least one even alternating path:

$$s_0 - f_1 - s_1 - f_2 \cdots f_k - s_k$$

with $(f_i, s_{i-1}) \in y^*$ and $(f_i, s_i) \in x^*$. By construction slot s_0 is uncovered in x^* and so by the optimality of x^* we have that $s_i < s_0$ for all $1 \leq i \leq k$, since otherwise x^* could be improved by by changing a flight assignment $(f_i, s_i), s_i > s_0$ to (f_i, s_0). To see this, we observe first that $s_1 < s_0$ (otherwise x^* could be improved by allocating f_1 to s_0). Now consider s_2. If $s_2 > s_0$, x^* could be improved by assigning f_2 to s_0 (note that this assignment is feasible since $s_1 < s_0$ and f_2 is assigned to s_1 in y^*). Thus, we also have $s_2 < s_0$. Repeating this argument yields the result for all $1 \leq i \leq k$. However, the fact that $s_k < s_0$ implies that y^* is not an optimal solution, which contradicts our assumption. Thus, x^* is a delay-minimizing solution.

A somewhat similar argument shows that any delay-minimizing solution uses the same slots. To see this, let x^* and y^* be any two delay-minimizing solutions. If these solutions do not use the same slots, the graph induced by the symmetric difference $x^* \oplus y^*$ must include at least one even alternating path with one end-node s_0 used in x^* and one end-node s_k used in y^*. But this implies that one of solutions doesn't minimize delay, leading to a contradiction. $\square$

This proposition implies that delay minimization is achieved under fairly mild conditions and that, in typical cases, there are a wide range of delay-minimizing solutions. As a consequence, the condition that flight-slot assignments should minimize overall delay requires little explicit consideration and there is much room to consider other criteria, e.g. equity, in addition to delay minimization.

3.2 Equity Principles

The notion of equity arises at two occasions during GDP procedures under CDM. It first appears in the initial allocation of slots to airlines at the start of a GDP, which is performed be the Ration-by-Schedule procedure. In addition, the issue of equity occurs in the inter-airline exchange of slots, which is done by the Compression procedure. The RBS algorithm aims to allocate slots among airlines in an equitable manner based on the notion that airlines have claims to slots based on their original schedules. The compression algorithm on the other hand aims to allocate slots by repeatedly moving up flights into open slots, based on the principle that airlines should be rewarded for slots they release. Both of these procedures resulted from extensive "war-gaming" sessions involving airline and FAA traffic flow managers. As such it is felt that they embody consensus concepts for fair allocation within this setting. In spite of their intuitive appeal however, the "hard coding" of these equity concepts into the actual procedures limits their applicability. It is not straightforward, for instance, to generalize the GDP approach to more complex settings such as the allocation of resources in en-route airspace environment or even to enhance the current procedures. Hence, it is desirable to separate the concept of equity from the actual algorithms.

Concerns about equity arise in many situations where scarce resources have to be allocated, ranging from the allocation of students to dorms and the allocation of kidneys to patients to the apportionment of house seats. Usually, the interpretation of what constitutes an equitable allocation depends to a large extent on contextual details: what is deemed fair in one environment may be undesirable in another. Nevertheless, the are certain common principles that return in many real-world problems and that may provide a constructive basis for reasoning about equity (we refer to Young (1994) for a comprehensive overview). As discussed in Young (1994), principles of equity are commonly based on pairwise comparisons, that is, allocations are evaluated by comparing pairs of claimants w.r.t their allocated portions. An allocation is equitable if no transfer of the resources is "justified". Informally, a justified transfer is a transfer from a less deserving claimant to more deserving claimant that is such that the less deserving claims remains so after the transfer. Of course, the critical aspect in this definition is the specification of who is less or more deserving. This is usually formalized through a so-called standard of comparison, which defines the priorities claimants have over allocations and may be context dependent.

The resulting concepts of equity are closely related to the use of *lexicographic minimax* objective functions in multi-objective optimization models. Minimax objective functions are commonly used in resource allocation problems where it is desirable to allocate limited resources equitably among competing activities (see Ignizio, 1982, Luss, 1999). Each activity has its own performance function, which typically measures the shortfall with respect to a specified target or goal. A minimax objective function minimizes the maximum performance function value among all activities (e.g. the maximum shortfall with respect to the targets). Models with a minimax objective function attempt to allocate limited resources equitably among the worst-off activities, which may however may not be sufficient as they leave open many possibilities for allocating resources among the activities that are not among the worst-off. To address this issue, a lexicographic minimax objective function is oftentimes used. A lexicographic minimax solution allocates resources in such a way that no performance function can be improved without worsening the performance of an activity that is already equal or larger. In this sense, the lexicographic minimax criterion closely corresponds to the absence of justified transfers criterion described above. In the following sections, we present models which are equivalent or nearly equivalent to both RBS and compression based on lexicographic minimax objective functions.

3.3 Optimization-based Slot Allocation

In this section, we show that the RBS and accrued-delay solutions are, in fact, minimax solutions to the assignment problem defined previously. We further show that a special case of the Optiflow objective function finds such a solution.

To formally state the equivalence, we let T represent the maximum delay allocated by the RBS algorithm, and define for each i, $0 \leq i \leq T$, the performance function

$$d_i = \mid \{f \in \mathcal{F} : cta_f - oag_f = i\} \mid .$$

We now can state the desired result.

Theorem 3.2. *The flight-slot assignment obtained by the RBS algorithm, the accrued-delay algorithm and the OPTIFLOW model, with $w_f = 1$ and $I(f) = oag_f$ for all $f \in \mathcal{F}$, lexicographically minimize the maximum delay with respect to the original flight schedule; that is, the allocation obtained by each of the three approaches lexicographically minimizes the vector*

$$d = (d_T, \ldots, d_0)$$

over all possible flight-slot allocations.

Proof. We assume w.l.o.g. that all oag times are different. The proof follows by considering the following propositions:

- *RBS algorithm ⇔ Accrued-delay algorithm.*
  Consider the schedules obtained by the respective algorithms. Let s be the first slot in which they differ, and let f be assigned to s under the accrued-delay algorithm and f' be assigned to s under the RBS algorithm. Under the logic of the accrued-delay algorithm, this would imply that $oag_{f'} < oag_f$. But then f' would precede f in RBS's queue, which would imply that f would have been assigned to s under RBS. Hence, both algorithms will results in the same schedule.

- *The assignment obtained by the accrued-delay algorithm lexicographically minimizes the maximum delay.*
  This follows by a sequential exchange argument. Let A_1 be a lexicographical min-max assignment and A_2 an assignment generated by the accrued-delay algorithm. We now will argue that A_1 and A_2 necessarily assign the same flight to the first slot. Suppose this is not the case so that flight f occupies the first slot, s_1, in A_2, but slot $s_k > s_1$ in A_1, and let g be the flight assigned to s_1 in A_1. It follows from the basic properties of the accrued delay algorithm that $oag_f < oag_g$, which implies Max$\{s_1 - oag_f, s_k - oag_g\}$ <Max$\{s_1 - oag_g, s_k - oag_f\}$. It then follows that the lexicographical min-max objective function can be improved for $A1$ by interchanging f and g. This is a contradiction to the optimality of $A1$ Repeating this argument for slots $2, \ldots, n$ yields the desired result.

- *The assignment obtained by the accrued-delay algorithm is optimal w.r.t. the OPTIFLOW model.*
  This follows by a similar exchange argument. Let A_1 be an optimal solution to the OPTIFLOW model and A_2 an assignment generated by the accrued-delay algorithm. Suppose this is not the case so that flight f occupies the first slot, s_1, in A_2, but slot $s_k > s_1$ in A_1, and let g be the flight assigned to s_1 in A_1. However, this would imply that interchanging the assignment of f and g would improve the OPTIFLOW objective function (since $oag_f < oag_g$), which is a contradiction. Again, repetition of the argument yields the desired result.

□

In other words, the allocation obtained by all three procedures is such that we cannot reduce a flight's allocated delay, d, without increasing the delay of another flight to a value larger than d.

Since the lexicographic minimax criterion is equivalent to the RBS procedure, which achieved acceptance after significant negotiations and "war-gaming" activities, it should be considered as the basis for allocation in other contexts. In particular, using a lexicographical minimax objective might prove especially useful if the allocation involves more complex combinations of resources (e.g., a region of airspace) and that require the solution of a more complex optimization problem.

4 Acknowledgments

This work was supported by NEXTOR, 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] Ball, M., R. Dahl, L. Stone, and T. Thompson, 1993, "OPTIFLOW Build-I Design Document, Version 1.5," *Optiflow Project Report to the FAA*, December.

[2] Ball, M.O., R. Hoffman, W. Hall, and A. Muharremoglu, 1998, "Collaborative Decision Making in Air Traffic Management: A Preliminary Assessment", NEXTOR technical report RR-99-3.

[3] Ball, M.O., R. Hoffman, D. Knorr, J. Wetherly and M. Wambsganss, 2000, "Assessing The Benefits Of Collaborative Decision Making In Air Traffic Management", in Proceedings of 3rd USA/Europe air Traffic Management R&D Seminar.

[4] Ball, M.O., Hoffman, R., Odoni, A. and Rifkin, R., 2000, "Efficient Solution of a Stochastic Ground Holding Problem", *Working Paper.*

[5] Bertsimas, D. ans S. Stock, 1998, "The Air Traffic Flow Management Problem with Enroute Capacities," *Operations Research*, 46:3, 406-422.

[6] Hoffman, R. and Ball, M.O., 2000, "A Comparison of Formulations for the Single Airport Ground Holding Problem with Banking Constraints", *Operations Research*, 48, 578-590.

[7] Hoffman, R., W. Hall, M. Ball, A. Odoni, and M. Wambsganns, 2000, "Collaborative Decision Making in Air Traffic Flow Management," *Working Paper.*

[8] Howard, K., Informal Memorandum to CDM Collaborative Routing Subgroup.

[9] Iginizio, J.P., 1982, "Linear Programming in Single- and Multiple-Objective Systems," , Prentice-Hall.

[10] Luss, H., 1999, "On Equitable Resource Allocation Problems: A Lexicographic Minimax Approach," *Operations Research*, 47:3, 461-378.

[11] Odoni, A.R., 1987, "The Flow Management Problem in Air Traffic Control," In *Flow Control of Congested Networks*, A.R. Odoni, L. Bianco, and G. Szego, eds., 269-288, Springer-Verlag, Berlin.

[12] Richetta, O. and A.R. Odoni, 1993, "Solving Optimally the Static Ground Holding Policy Problem in Air Traffic Control," *Transportation Science*, 24, 228-238.

[13] Young, H.P., 1994, "Equity in Theory and Practice," , Princeton University Press.

[14] Wambsganns, M., 1996, "Collaborative Decision Making Through Dynamic Information Transfer," *Air Traffic Control Quarterly*, 4, 107-123.