Optimization and Mediated Bartering Models for Ground Delay Programs

Thomas Vossen and Michael Ball
Robert H. Smith School of Business
and Institute for Systems Research
University of Maryland
College Park, MD 20742 USA

August 23, 2001

Abstract

The Federal Aviation Administration (FAA) and the airline community within the US have recently adopted a new paradigm for air traffic flow management, called Collaborative Decision Making (CDM). A principal goal of CDM is shared decision making responsibility between the FAA and airlines, so as to allow airlines to control decisions that involve economic tradeoffs. So far, CDM has primarily led to enhancements in the implementation of Ground Delay Programs, by changing procedures for allocating slots to airlines and for exchanging slots between airlines. In this paper, we discuss how these procedures may be formalized through appropriately defined optimization models. In addition, we describe how inter-airline slot exchanges may be viewed as a bartering process, in which each "round" of bartering requires the solution of an optimization problem. We compare the resulting optimization problem with the current procedure for exchanging slots, and discuss the possibilities for increased decision making capabilities by the airlines.

1 Introduction

Recently, the U.S. Federal Aviation Administration (FAA) and the major U.S. airlines have embraced a new initiative to improve Air Traffic Flow Management. This initiative, called *Collaborative Decision Making* (CDM), is based on the recognition that improved data exchange and communication between the FAA and the airlines will lead to better decision making (Wambsganss, 96). In particular, the CDM philosophy emphasizes that decisions with a potential economic impact on airlines should be decentralized and made in collaboration with the airlines whenever possible. While the CDM paradigm encompasses a wide range of applications in Air Traffic Flow Management, its primary focus so far has been the implementation and enhancement of Ground Delay Programs (GDPs). GDPs are issued by the FAA as a response to predicted periods of significant capacity-demand imbalance at an airport. Such an imbalance is typically caused by a decline in airport arrival capacity, usually due to adverse weather conditions. In a GDP, the number of flights that are scheduled to arrive during the expected periods of congestion is reduced to the level of the reduced arrival capacity. This is done by assigning in-bound flights delays prior to their departure; hence the term ground delay. The motivation behind a GDP is that it is both safer, and less expensive, to delay a flight on the ground (i.e. before its take-off) than to delay a flight that is airborne. A key issue in implementing a GDP is the allocation of delay to the in-bound flights. This defines a resource allocation problem, in which available arrival capacity is allocated to flights. Typically, arrival capacity is partitioned into a sequence of arrival slots, each of which can handle the arrival of a single flight. This problem is also known as the Ground Holding Problem, and has been studied extensively. The basic version of the problem, in which both demand and airport capacity are assumed to be deterministic, was first systematically described in (Odoni, 1987). More advanced models have, among others, addressed stochastic arrival capacity (Ball et al., 2000, and Richetta, 1993), the incorporation of banking constraints (Hoffman and Ball, 1998), and the extension to en-route capacities (Bertsimas and Stock, 1998). A common feature of these approaches is that they aim to find allocations with minimum overall cost, based on a trade-off between the costs of ground delay and airborne delay. As such, these approaches can be said to follow a central planning paradigm, in that system-wide optimal solutions are developed without considering the effects on a single airline.

As a consequence, these models are not readily applicable under the CDM paradigm, and it is therefore no surprise that the GDP enhancements implemented under CDM follow quite a different approach. The number of enhancements that have recently been implemented are numerous: examples include improved data-exchange, better situational awareness tools, and increased flexibility for the airlines (see Ball et al., 1998, 2000). However, the most significant change with regard to the allocation of resources has come through the consensus recognition that airlines have claims on the available arrival capacity, based on their original flight schedules. This realization has led to a key change in the way arrival capacity is allocated, and to the introduction of a procedure for inter-airline slot exchange. Under CDM, arrival capacity is allocated to the airlines by a procedure called *Ration-By-Schedule* (RBS). The main purpose of this procedure has been the removal of disincentives for airlines to provide accurate information about delays and cancellations. The procedure for inter-airline slot exchange under CDM is called *Compression.* This procedure seeks to maximize utilization of the available arrival capacity in the presence of delays and cancellations, and attempts to do so in a fair and equitable manner. Though the RBS and Compression procedures are intuitively appealing and highly transparent, they also have a limited scope. Consequently it is difficult to incorporate more advanced trade-offs into the decision processes in a GDP, such as those proposed in various extensions of the Ground Holding Problem.

The main objective in this paper is to formalize and extend the inter-airline slot exchange procedure (compression) currently used in GDPs. Our main result is to demonstrate how the current procedure for exchanging slots corresponds to an appropriately defined optimization model. More specifically, we describe how the initial allocation of slots and the inter-airline slot exchange may be interpreted as an assignment problem with an appropriately defined lexicographic minimax objectives. The resulting optimization model has several advantages. First, the minimax objective may be used in more complex versions of the ground holding problem that were discussed before. Second, the resulting optimization model may serve as basis for enriching the trade-offs available to the airlines. By demonstrating that the criterion of lexicographically minimizing the maximum delay applies to the ration-by-schedule algorithm, the compression algorithm as well as an emerging method for allocating enroute resources, we show that this criterion embodies a fundamental notion of fairness that has emerged from a variety of FAA-airline gaming excercises and negotiations over the past few years. Although the optimization model can be directly motivated, we further show the the inter-airline slot exchange may be viewed as a mediated bartering process and that the optimization model represents the mediator's problem. We compare the resulting optimization problem with the current procedure for exchanging slots, and discuss the possibilities for increased decision making capabilities by the airlines.

This paper is organized as follows. Section 2 provides a background on GDPs, in particular on the procedures used under CDM. Section 3 discusses the optimization-based approach to the GDP procedures. In Section 4, we discuss how the inter-airline exchange of slots may be interpreted as a form of bartering. Finally, Section 5 provides conclusions and avenues for further research.

2 GDP Procedures

The FAA continuously monitors airports throughout the US for capacity-demand inequities. Whenever it is predicted that the number of flights arriving at an airport within a 15-minute interval exceeds the number of flights scheduled to land, the FAA is by law forced to respond. Short periods of congestion are usually resolved by airborne tactics, such as re-routing and variations in airborne speed. GDPs on the other hand primarily address longer periods of congestion at an airport. Usually, a GDP spans a period of 4 hours or longer, and is initiated 3-4 hours in advance.

The implementation of a GDP involves a large number of considerations, such as determining which flights will be affected, deciding the program's duration, and estimating the available capacity (available arrival capacity is partitioned into a sequence of arrival slots, each of which can handle one flight). Based on these factors, a GDP will result in an assignment of the affected flights to the available arrival slots. Once a flight has been assigned an arrival slot, its adjusted departure time (and hence the amount of ground delay) is easily derived, since en-route travel times can be predicted with reasonable accuracy. To maximize utilization of available arrival capacity, the assignment of flights to slots may have to be updated frequently during the course of a GDP. Partly, this is due to inherent uncertainty of flight delays (e.g. flights may experience mechanical delays, etc.). Moreover, airlines are allowed to rearrange their "portion" of the schedule. To mitigate the adverse effects of a GDP on its operations, airlines oftentimes cancel flights and substitute flight-slot assignments. Hence, a GDP

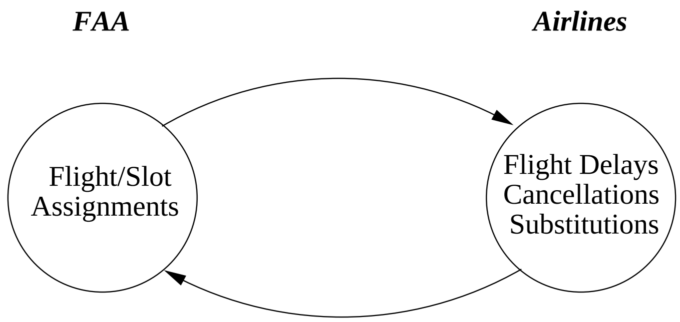

Figure 1: FAA/airline interaction in a GDP

may be considered a iterative process as depicted in Figure 1, in which the FAA repeatedly assigns flights to slots, based on airlines' flight updates, substitutions and cancellations.

Prior to the implementation of GDP enhancements under CDM, flights were assigned to slots by a first-come, first-served algorithm affectionately known as *Grover Jack*. The affected flights were ordered according to their most recent estimated arrival times, so the net effect of the Grover Jack algorithm was more or less to stretch out the incoming flights over time. Though intuitively appealing, this method of assigning flights to slots actually penalized the airlines for providing accurate information. This effect, known as the double penalty issue, is perhaps best explained by considering the following example. Suppose a flight is estimated to arrive at 10:00. Suppose now that the flight is delayed for 30 minutes due to mechanical problems, so that its estimated arrival time will change to 10:30. If a GDP is implemented in which the flight is delayed for another 30 minutes, its total delay will be 60 minutes. However, had the airline not notified the FAA of its mechanical delay, it would have only been assigned a delay of 30 minutes ! As a consequence, airlines usually wouldn't provide the FAA with accurate estimates of their flight delays. In addition, airlines usually wouldn't report flight cancellations in a timely manner, as their was no incentive to do so. As a result, GDP decisions were based on poor data, which led to inefficient programs.

The GDP enhancements implemented under CDM address these issues by instituting a fundamental change in the allocation of capacity by the FAA. Instead of assigning flights to slots, the CDM "philosophy" views the allocation of capacity essentially as assigning slots to airlines. This has led to the introduction of two new allocation mechanisms, called *Ration-By-Schedule* (RBS) and *Compression*. The RBS algorithm creates an initial allocation of slots to airlines, based on the consensus recognition that airlines have claims on the available arrival capacity through the original flight schedules. Once arrival slots have been allocated, the Compression algorithm performs schedule updates by an inter-airline slot exchange, which aims to provide airlines with an incentive to report flight cancellations and delays. Hence, under CDM the interaction between the

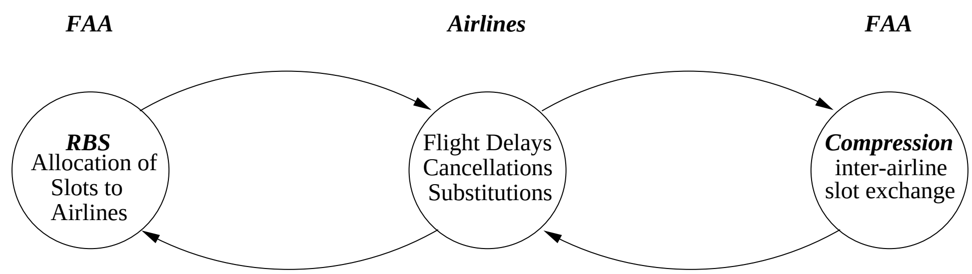

Figure 2: FAA/airline interaction in a GDP under CDM

FAA and the airlines may be represented as shown in Figure 2. 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 $O : \mathcal{F} \rightarrow \mathcal{A}$ 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 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.

To illustrate the compression algorithm, let us consider the example shown in Figure 3. The leftmost figure represents the flight-slot assignment prior to the execution of the compression algorithm. Associated with each flight is an earliest time of arrival, and each slot has an associated slot time. Note that there is one canceled flight. The rightmost figure shows the flight schedule after execution of the compression algorithm: as a first step, the algorithm attempts to fill AAL's open slot. Since, there is no flight from AAL that can use the slot, the slot is allocated to UAL, and the process is repeated with the next open slot.

(a) Initial Assignment (b) Compression Assignment

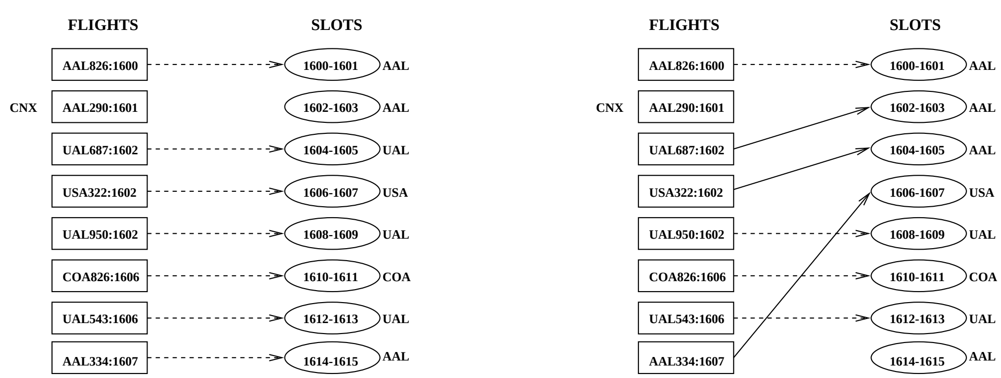

Figure 3: Compression Example.

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

The Linear Programming formulation of the problem is as follows.

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

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 $1 \le i \le k$, 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 $1 \le i \le k$. 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. □

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

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

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.

3.4 Optimization-based Slot Exchange

The inter-airline exchange of slots in GDPs is performed by the Compression algorithm. As we saw earlier the compression algorithm repeatedly fills "holes" in the schedule, giving priority to flights from the airline that owns the open slot. In this section, we represent the inter-airline exchange of slots as an assignment problem with an appropriately defined lexicographic minimax objective. In the next section, we shall see that this model closely captures the inter-airline slot exchange achieved by compression.

In order to specify a lexicographic minimax objective that captures the exchange of slots in compression, it is first necessary to define a set of performance functions with appropriate targets or goals associated with each performance function. These have to be such that the resulting lexicographic minimax solution incorporates the notion of "paying back" airlines for the slots they release. To do this, we associate each (non-canceled) flight with a *goal* slot, and use as a performance function the shortfall from this goal slot (that is, the difference between the assigned slot and the goal slot). A goal slot is defined for each non-canceled flight by specifying a mapping

- $g : \mathcal{F}/\mathcal{C} \rightarrow \mathcal{S}$, such that $g(f) \in I(\mathcal{F}_a)$ when $O(f) = a$,

that is, each flight's goal slot should be owned by that flight's airline.

The idea behind the use of goal slots is that in a "completely fair" solution each airline should be able to use each of the slots that it owns. Because of schedule disruption, it may be unable to do so. When a goal slot is not attainable for a flight, minimization of the deviation with respect to the goal embodies the concept of minimizing the deviation from the most fair allocation. To clarify this concept, let us consider the example depicted in Figure 4.

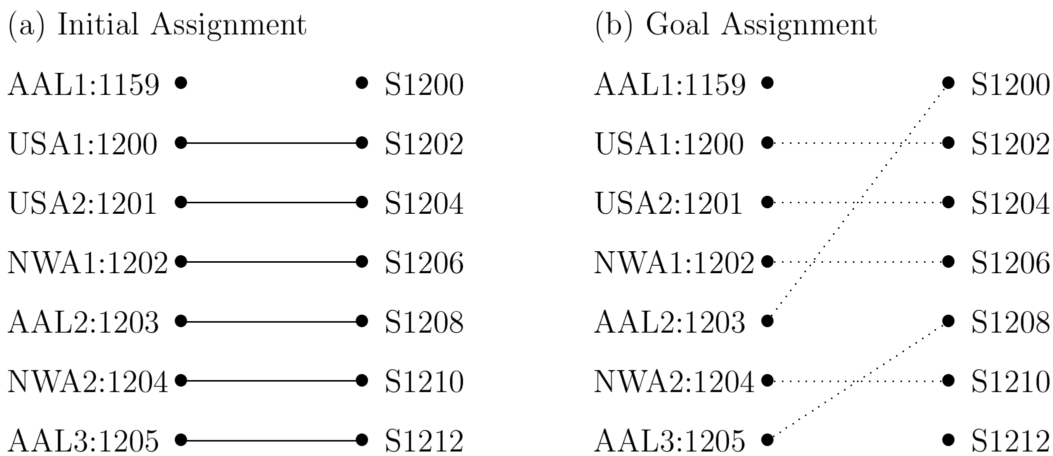

Figure 4: Example of initial allocation and assignment of goals.

In the initial assignment depicted in Figure 4, AAL has cancelled its first flight, leading to an open slot. Ideally, it would assign its next flight, AAL2, to this slot. Thus, slot S1200 represents the goal for flight AAL2 (note that AAL2's earliest arrival time is later than 12:00). Subsequently, AAL would like to assign AAL3 to slot S1208, which is now vacated by AAL2. Thus, slot S1208 represents the goal for flight AAL3.

In general, there may be many ways in which an airline could assign goals to flights. A natural requirement would be that each flight is assigned one goal slot, and that each slot can be used at most once as a goal. This latter requirement is appropriate since the model will seek to minimize the deviation between the flight and its assigned goal. Assigning two flights to the same goal would be analogous to assigning two flights to a single landing time slot. Furthermore, if an airline has k flights, it is clear that it should use as goals its first (earliest) k slots. A natural "default" assignment of flights to goals would be achieved by ordering the flights according to increasing earliest time-of-arrival (ETA) and goals according increasing to slot times and then assigning the 1st flight to the 1st goal, 2nd flight to the 2nd goal, etc. In the experiments, we later describe we use this approach. On the other hand, the ability to order, and thus prioritize, flights could enable a higher degree of airline control in the exchange of slots. We will discuss this issue later in this section.

Suppose now we have determined a flight-goal assignment, represented by g. Our next step is to determine an objective function for the OPTIFLOW model that lexicographically minimizes each flight's deviation from its goal slot. To do so, we specify the objective functions as follows:

$$D_k(x) = \sum_{i \in \mathcal{F}, j \in \mathcal{S}: j - g(i) = k} x_{i,j},$$

for $1 \leq k \leq k_{\max}$, where $k_{\max}$ represents the maximum goal deviation. In other words, $D_k(x)$ represents the number of flights with a goal deviation of k. Since our aim is to lexicographically minimize the maximum deviation, these objective functions are ordered in such a way that the higher the goal deviation, the higher the priority. As such, our objective is to lexicographically minimize the vector

$$(D_{k_{\max}}(x), \ldots, D_1(x)). \tag{1}$$

As discussed in Ignizio (1982), this can be done by using an appropriate weighted sum, which leads to an objective function

$$\text{MIN} \sum_{1 \leq k \leq k_{\max}} C_k D_k(x)$$

with $C_{k_{\max}} \gg \cdots \gg C_1$. These weights can be represented as $C_k = kC$, with C a sufficiently large constant. Thus, we can rewrite the objective function as

$$\text{MIN} \sum_{f \in \mathcal{F}, s \in \mathcal{S}: s > g(f)} C(s - g(f)) x_{f,s}.$$

While the OPTIFLOW model appended with this objective function will lexicographically minimize the maximum goal deviation, it will not necessarily minimize overall delay. To ensure that overall delay is minimized, we furthermore combine this objective with the "tardiness"coefficient that was used in the OPTIFLOW model. Thus, the overall objective function coefficients are defined as

$$c_{f,s} = \begin{cases} C(s - g(f)) + (s - g(f))^{1+\epsilon} & \text{if} \quad s > g, \\ (s - g(f))^{1+\epsilon} & \text{if} \quad s \leq g, \end{cases} \tag{2}$$

with $C \gg 1$ and $0 < \epsilon \ll 1$.

Theorem 3.3. *The OPTIFLOW model using objective function (2) yields a solution which lexicographically minimizes (1) and also minimizes total delay.*

Proof. Lexicographic minimization follows by construction from the first term in the objective function (e.g. for $s > g$). Minimization of total delay follow from including the second term (and case $s \leq g$) together with Theorem 3.1. □

This result provides the basis for using an assignment model in place of the compression algorithm.

3.5 Comparison

In this section, we compare the performance of the optimization model for inter-airline slot exchange with the Compression algorithm. First, we compare their resulting flight schedules on a small number of real-world ground delay program scenarios. This provides a more aggregate view of the differences and similarities between both models. Next, we analyze the differences between the optimization based approach and the compression algorithm through a number of smaller examples. This serves to illustrate the advantages and disadvantages of using the optimization model versus the compression algorithm.

3.6 Experimental Results

We compared the performance of the optimization model with the Compression algorithm on four scenarios, which were derived from real-world GDPs. Three of the data sets that were used represented GDPs at Newark International Airport (EWR), while one of the data sets considered a GDP at Los Angeles International Airport (LAX). The data gathered for each compression scenario consisted of the flights and slots in the GDP, the initial assignment of flights to slots, the earliest arrival times for each flight, and the set of flights that were canceled. The four scenarios are summarized in Table 1. For each of the scenarios we ran both the optimization model and the Compression algorithm. The results are shown in Tables 2 through 5. The tables show for both the absolute and the relative delay savings for each airline (delay savings are measured in minutes). In addition, the tables show for each airline the *baseline savings*, that is, the reduction in delay each airline would have been able to achieve by itself. Baseline savings provide a convenient basis for comparison of delay reduction on an airline-by-airline basis. An airline will always achieve this amount of delay savings, and the fact that more total savings are possible is exactly due to inter-airline slot swapping. The results in Tables 2 through 5 indicate that the optimization model results in flight-slot assignments that are very similar to those obtained by the Compression algorithm. Hence, it appears that the use of targets in the lexicographic minimax objective closely captures the notion of "paying back" from compression.

Table 1: Problem Characteristics

	EWR 01/01/96(1)	EWR 01/01/96(2)	EWR 01/02/96	LAX 01/01/97
Number of Flights	73	94	54	62
Number of Cancellations	12	21	6	10

Table 2: Delay reduction for Scenario EWR, 01/01/96(1)

Airlines	Comp Absolute	Comp Relative	Opt Absolute	Opt Relative	Baseline Absolute	Baseline Relative
COA	402	46.53	406	46.99	281	57.00
UAL	200	23.15	195	22.57	142	28.80
TWA	17	1.97	17	1.97	0	0.0
AAL	123	14.24	126	14.58	70	14.20
ACA	2	0.23	0	0.00	0	0.0
USA	38	4.40	38	4.40	0	0.0
BSK	2	0.23	2	0.23	0	0.0
NWA	19	2.20	19	2.20	0	0.0
AWE	14	1.62	14	1.62	0	0.0
DAL	19	2.20	19	2.20	0	0.0
KMR	3	0.35	3	0.35	0	0.0
CAA	0	0.0	0	0.00	0	0.0
LOT	2	0.23	2	0.23	0	0.0
SJI	10	1.16	10	1.16	0	0.0
COM	13	1.50	13	1.50	0	0.0
TOTAL	864	100.00	864	100.00	0	0.0

Table 3: Delay reduction for Scenario EWR, 01/01/96(2)

Airlines	Comp Absolute	Comp Relative	Opt Absolute	Opt Relative	Baseline Absolute	Baseline Relative
FDX	0	0.0	0	0.00	0	0.0
COA	521	50.63	524	50.92	420	75.0
NWA	79	7.68	77	7.48	0	0.0
ACA	4	0.39	4	0.39	0	0.0
UAL	171	16.62	168	16.33	68	12.14
AAL	81	7.87	81	7.87	72	12.86
USA	84	8.16	87	8.45	0	0.0
DAL	29	2.82	29	2.82	0	0.0
DLH	0	0.0	0	0.00	0	0.0
TWA	2	0.19	2	0.19	0	0.0
BSK	6	0.58	5	0.49	0	0.0
AWE	6	0.58	6	0.58	0	0.0
BAW	6	0.58	6	0.58	0	0.0
KMR	16	1.55	16	1.55	0	0.0
LOT	24	2.33	24	2.33	0	0.0
TOTAL	1029	100.00	1029	100.00	560	100.00

Table 4: Delay reduction for Scenario EWR, 01/02/96

Airlines	Comp Absolute	Comp Relative	Opt Absolute	Opt Relative	Baseline Absolute	Baseline Relative
COA	231	64.71	270	75.63	167	85.20
ACA	40	11.20	10	2.80	0	0.0
SJI	3	0.84	3	0.84	0	0.0
COM	2	0.56	2	0.56	0	0.0
N4I	2	0.56	2	0.56	0	0.0
UAL	60	16.81	60	16.81	29	14.80
MXA	2	0.56	0	0.00	0	0.0
NWA	5	1.40	0	0.00	0	0.0
VIR	3	0.84	3	0.84	0	0.0
TWA	3	0.84	3	0.84	0	0.0
PAL	2	0.56	0	0.00	0	0.0
AJM	1	0.28	1	0.28	0	0.0
USA	1	0.28	1	0.28	0	0.0
AAL	1	0.28	1	0.28	0	0.0
CAA	1	0.28	1	0.28	0	0.0
TOTAL	357	100.00	357	100.00	196	100.0

Table 5: Delay reduction for Scenario LAX, 01/01/97

Airlines	Comp Absolute	Comp Relative	Opt Absolute	Opt Relative	Baseline Absolute	Baseline Relative
UAL	153	42.62	142	39.55	127	53.59
AAL	72	20.06	66	18.38	70	29.54
SWA	25	6.96	32	8.91	18	7.59
TWA	38	10.58	38	10.58	0	0.0
ASA	6	1.67	6	1.67	0	0.0
SER	0	0.0	0	0.00	0	0.0
DAL	8	2.23	8	2.23	0	0.0
FDX	4	1.11	4	1.11	0	0.0
RKT	2	0.56	4	1.11	0	0.0
ROA	9	2.51	9	2.51	0	0.0
AMX	2	0.56	8	2.23	0	0.0
ANZ	2	0.56	4	1.11	0	0.0
AWE	0	0.0	0	0.00	0	0.0
USA	24	6.69	24	6.69	22	9.28
COA	2	0.56	2	0.56	0	0.0
NWA	6	1.67	6	1.67	0	0.0
FFT	6	1.67	6	1.67	0	0.0
TOTAL	359	100.00	359	100.00	237	100.00

3.7 Analysis

Even though the resulting schedules obtained by the compression algorithm and the optimization model were shown to be very similar, they were never identical. Here, we show that this is in fact unavoidable, that is, we provide an example in which for *any* assignment of goal slots to flights, the optimization model may find a solution that is different from the solution found by the compression algorithm. Next, we provide an example that shows one of the main reasons for this discrepancy.

3.7.1 Difference between Compression and Optimization model

To show that it is generally impossible to find a solution that mimics compression, we use the following counterexample. The example starts with an initial schedule as shown in Figure 5, which also shows the results obtained by the Compression algorithm. In

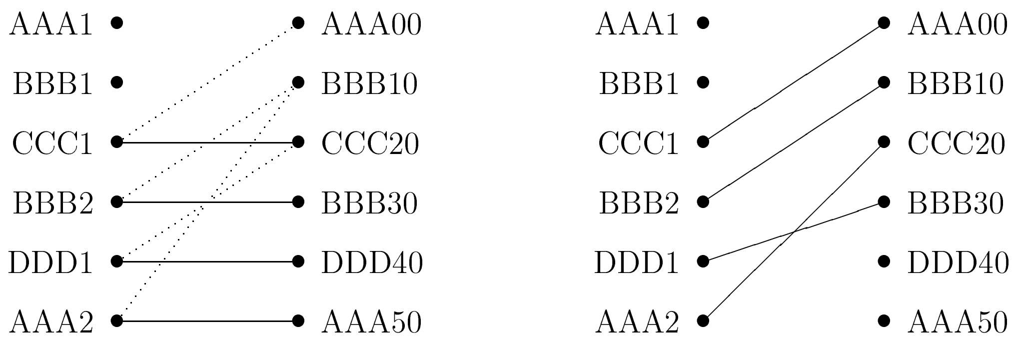

Figure 5: Initial Schedule and Compression Results. The solid lines represent flight-slot assignments, the dashed lines the earliest arrival times.

the compression algorithm, first flight CCC1 will moved up to slot AAA00 and flight AAA2 to slot CCC20. Secondly, flight BBB2 will be moved up to slot BBB10 and flight DDD1 to BBB30.

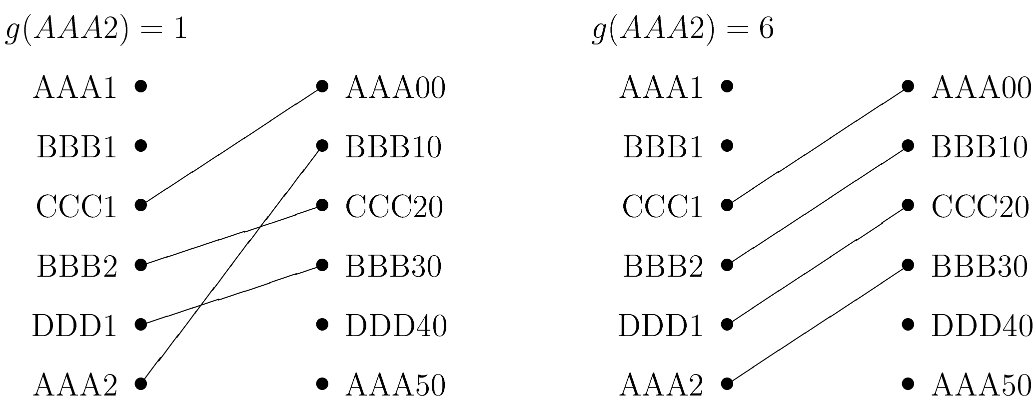

Figure 6: Possible optimization model solutions.

We now show that it is impossible to obtain this flight-slot assignment by using the optimization model. To do this, we consider all possible ways of assigning goals to the flights. Since airlines CCC and DDD own only one slot, their flights necessarily have these slots as their goals, i.e. $g(CCC1) = CCC20$ and $g(DDD1) = DDD40$. Flight AAA2 can be assigned either slot $AAA00$ or slot $AAA50$ as its goal, and similarly flight BBB2 can have either slot $BBB10$ or slot $BBB30$ as its goal. Thus, overall we have four possible ways of assigning goals to flights. However, it is easily seen that the resulting solution is independent of the goal of flight $BBB2$. Consequently, there are two possible solutions, dependent on whether $g(AAA2) = 1$ or $g(AAA2) = 6$. These solutions are shown in Figure 6. The example shows that it is impossible to assign goals to flights in such a way that the optimization model will yield the same result as the compression algorithm.

3.7.2 Order dependence in the Compression Algorithm

The previous example showed that the optimization model cannot be expected to yield exactly the same results as compression. Here, we provide an example that illustrates one of main reasons for the differences observed in the previous section, i.e. the influence of the order in which slots are compressed in the compression algorithm. The example considers an initial situation as depicted in Figure 7.

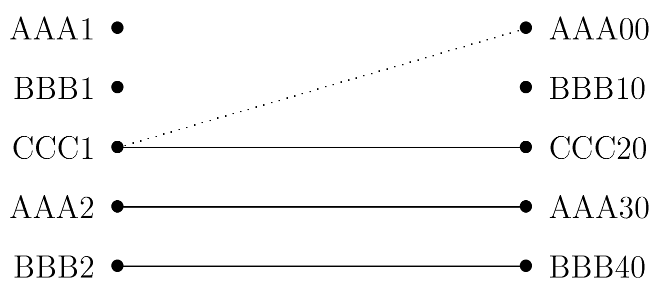

Figure 7: Initial schedule. The solid lines represent flight-slot assignments, the dashed lines the earliest arrival times.

Suppose now the compression algorithm fills the open slots in the order AAA1, BBB1. Compressing slot AAA1 will result in the assignment of slot AAA00 to flight CCC1, slot CCC20 to AAA2, and slot AAA30 to BBB2. After this, compressing slot BBB1 will not change any flight-slot assignments. The resulting flight-slot assignment is depicted in Figure 8(a). However, if the open slots are compressed in the reverse order, that is, first BBB1 and then AAA1, the result will be as shown in Figure 8(b).

We observe that the order of flights AAA2 and BBB2 is reversed, dependent on the order in which slots are compressed. The optimization model, on the other hand, will obtain the the flight-slot assignment that corresponds to compressing slot AAA1 first.

3.8 Discussion

The previous sections showed that appropriately defined lexicographic minimax objectives closely capture the fairness trade-offs used in both the initial alloaction of slots to airlines and in the inter-airline exchange of slots currently employed by the FAA. A key

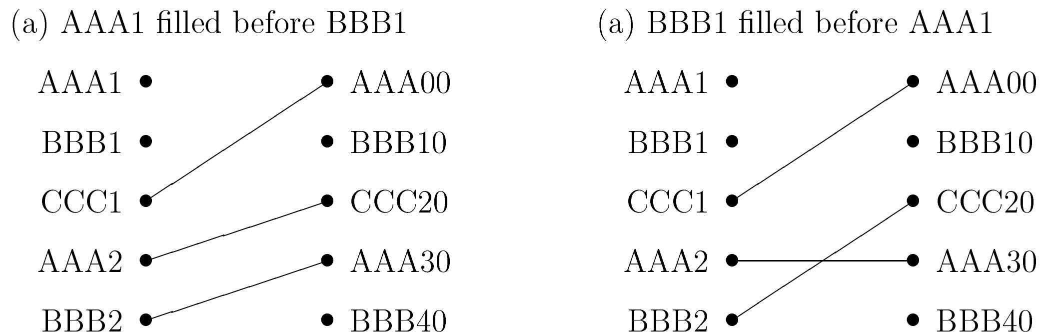

Figure 8: Compression results.

aspect in this approach is the use of *goal* slots, which had to be satisfied by an airline's flights. One advantage of using goal-setting in our optimization models could be that it leads to more flexible input requirements. Specifically, the use of goal-setting would allow less specific inputs from the airlines. For example, one scenario could be that airlines simply submit an ordered flight list; from these lists, an airline's goals could be determined. Moreover, it might also be possible to leave the assignment of goals to flights up to the airlines, thus providing them an additional means of control and flexibility. It should be noted however that such an approach would require further analysis of its fairness aspects.

Perhaps the biggest advantage of the optimization-based approach is the possibility to incorporate the use of goal-setting together with a lexicographic minimax objective into more complex settings. For example, the objective function used in the optimization model could also be incorporated into optimization models for the Ground Holding Problem with banking constraints, which were proposed by Hoffman and Ball (1998). Under the CDM paradigm, this could provide the airlines added control over their flight schedules. In addition, our approach might be useful when considering the extension to en-route capacities (see Bertsimas and Stock, 1998). The complexity of this problem is such that it is unlikely that a simple method such as the compression algorithm could be used successfully.

4 A Revised View of Inter-Airline Slot Exchange

The GDP processes employed under CDM provide a basic mechanism for interaction between the airlines and the FAA. Initially, the FAA allocates slots using RBS. Through the resulting slot allocations, the FAA implicitly disseminates a set of constraints to each airline. Airlines may react to these by canceling flights and/or substituting flight-slot assignments, depending on their individual economic considerations and preferences. Finally, the airlines submit their preferences to the FAA, which uses Compression to achieve a final allocation that maximizes overall slot utilization. Currently, airlines primarily exercise control over their flight schedules by canceling flights and executing flight-slot substitutions. Their input into the slot exhange achieved by Compression is limited: given their preferences (which effectively are provided by the earliest arrival times of each flight), Compression follows a strict procedure for filling the "holes" in the schedule.

A relevant question therefore is how the delay savings are distributed among flights in the Compression algorithm and, in particular, whether airline control could be extended so that a more beneficial exchange of slots could be achieved. In the previous section, we briefly addressed this question by allowing airlines to assign goals to their flights, which the airlines could use to prioritize possible delay reductions among their flights. Here, we further address this issue by re-examining the role of the compression algorithm. To do this, we interpret the inter-airline exchange of slots in Compression as a form of bartering, in which the FAA acts as a "broker" matching offers proposed by the airlines. In particular, we discuss how the flight-slot assignment obtained by the optimization problem might be viewed as a "match" between slot-exchange offers proposed by the airlines. The advantage of this interpretation is that it allows us to analyze the role and limitations of user preferences under the current Compression algorithm, and that it indicates several possible extensions and increased flexibility for the airlines in the inter-airline exchange of slots.

In the remainder of this section we describe in greater detail how the inter-airline slot exchange might be considered a form of bartering between the airlines, with the FAA acting as a mediator. In addition, we discuss how the FAA's "mediation" problem may be viewed as an optimization problem, and in particular that a flight assignment obtained by the optimization problem described in the previous section may be interpreted as a solution to this mediation problem.

4.1 A Model for Mediated Bartering

We start by describing a general model of mediated bartering. In the next section we will describe its application to the air traffic management context. Consider a set of agents, $\mathcal{A}$, where each agent, $a \in \mathcal{A}$, owns a set of goods, $\mathcal{S}_a$. Each agent potentially desires to exchange one or more of its goods for goods owned by other agents. Here, we assume one-for-one exchanges, so that a single good owned by one agent is exchanged for a single good owned by another agent. The set of desired exchanges is characterized in terms of offers. For each good, $s \in \mathcal{S}_a$, we define an offer as the ordered pair (s, T_s) where T_s is a set of goods that agent a is willing to accept in exchange for s. To carry out the exchange process, there is a first stage in which each agent generates a set of offers and submits the entire set to a mediator. In the second stage the mediator determines the set of offers to accept. The mediator specifies both which offers to accept and for each accepted offer, (s, T_s), which element in T_s is exchanged for s. We note that a key aspect of the entire procedure is the criterion used by the mediator both to determine which offers to accept and to determine the exchange element for each offer. This criterion will influence (perhaps very strongly) the offers proposed by each agent.

We can represent the mediator's problem in terms of a directed network. The node set, N is the set of all goods and the arc set, A, is the set of possible exchange pairs, i.e.

$$\begin{aligned} N &= \bigcup_{a \in \mathcal{A}} \mathcal{S}_a \\ A &= \{(s,t) : s \in N \text{ and } t \in T_s\} \end{aligned}$$

We note that minimally feasible exchange sequences correspond to cycles in the network. For example, suppose that goods 1,2 and 3 with owners A, B and C, form a directed cycle. Then, an exchange is possible in which A transfers good 1 to B, B transfers good 2 to C and C transfers good 3 to A. Thus, the problem of finding a feasible set of exchange sequences is equivalent to finding a set of non-intersecting directed cycles in (N, A). The corresponding optimization problem can be formulated as an assignment problem and is, in fact, closely related to the assignment relaxation of the traveling salesman problem. To formulate an optimization model, we require a cost function, which we assume is given in terms of exchange costs, c_{st} for each ordered pair of goods $(s,t) \in A$. The problem can now be defined as:

$$\begin{aligned} \text{Min:} \quad & \textstyle\sum_{s \in N} \sum_{t \in T_s} c_{st} x_{st} && && (3) \\ \text{s.t.} \quad & \textstyle\sum_{t \in T_s} x_{st} + y_s = 1 && \text{for all } s \in N && (4) \\ & \textstyle\sum_{t : s \in T_t} x_{ts} + y_s = 1 && \text{for all } s \in N && (5) \\ & x_{st}, y_s \in \{0,1\} && \text{for all } (s,t) \in A \text{ and } s \in N && (6) \end{aligned}$$

The variable y_s is 0 if the offer (s, T_s) is accepted and is 1 if it is rejected. When $y_s = 0$, $x_{st_1} = 1$ for some $t_1 \in T_s$ where the owner of s receives t_1 in exchange for s. In addition, there is a t_2 for which $y_{t_2 s} = 1$ where the owner of t_2 receives s in exchange for t_2. Note that the same slack variable, y_s, appears both in constraint sets (4) and (5). This insures that if $y_s = 1$ then $x_{st} = 0$ for all t and $x_{t's} = 0$ for all t'.

We now generalize this model slightly to allow an agent greater control over its overall set of goods. Suppose an agent proposes two offers, (s_1, T_{s_1}) and (s_2, T_{s_2}), but has the following desire: that if the agent did not receive one of the goods in T_{s_1}, i.e. the offer is not accepted, then the agent would be assured of keeping good s_2 (another good owned by the agent). With this motivation, we define an offer more generally as a three tuple, $(s, T_s, \rho(s))$, where $\rho(s) \in \mathcal{S}_a$ and a is the owner of s. We interpret $\rho(s)$ as the good owned by a, which a wishes to insure it retains if the offer is not accepted. We require that the mapping $\rho(*)$ is one-to-one and define $\rho^-(t)$ as its inverse, i.e. $\rho^-(t) = \{s : \rho(s) = t\}$. The one-to-one requirement is necessary to insure that a single agent cannot cause the broker's problem to become infeasible by the manner in which that agent defines it offers. The new optimization model is obtained by replacing constraint (5) with

$$\textstyle\sum_{t : s \in T_t} x_{ts} + y_{\rho^-(s)} = 1 \quad \text{for all } s \in N \qquad (7)$$

Figure 9 illustrates one cycle in a solution obtained under this model. Here the "arc" from s_1 to $\rho(s_1) = s_2$ is chosen. Note that in this case, the agent relinquishes s_1, keeps s_2 and obtains a good in T_{s_2}. The use of the ρ function implements a type of hierarchical conditional exchange:

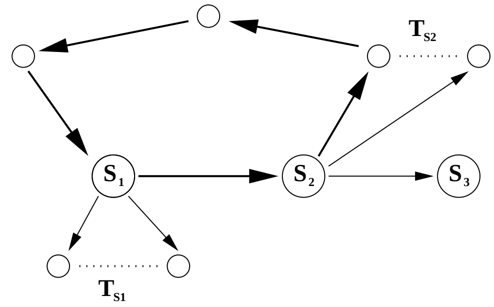

Figure 9: Cycle from Solution to Mediator's Problem Using "ρ-Arc"

option 1: s_1 exchanged for a good in T_{s_1} and s_2 exchanged for a good in T_{s_2}

option 2: s_2 retained and s_1 exchanged for a good in T_{s_2}

In fact, there is a complex recursive hierarchy of possible exchanges. For example, in the figure, $\rho(s_2) = s_3$ so that more complex conditional exchanges would result if the arc from s_2 to s_3 were also chosen. On the other hand, if for a given agent only "ρ"-arcs were chosen then this is equivalent to all of its offers being rejected.

Clearly a major component of this model, and its impact, is the definition of the objective function. Generally speaking the objective function will be application specific. Later in this section we define an objective function for the airline application.

4.2 Compression as Inter-Airline Bartering

We now consider the application of the model just described to the case of slot exchange among airlines. The goods are the slots available in a GDP; the set of agents, $\mathcal{A}$, is made up of the participating airlines; and $\mathcal{S}_a$ consists of those slots assigned to airline a by RBS. In the remainder of this section we illustrate how the Compression algorithm fits within the bartering framework described in the previous section. Furthermore, we will show that the assignment model formulated to solve the mediator's problem is identical to the assignment model proposed earlier in the paper to replace the Compression algorithm.

Given the initial allocation of slots (based on RBS), an airline could in principle make any offer to exchange slots. An exchange offer consists of a slot an airline is willing to give up and a set of possible slots it would be willing to accept in return.

Let us examine why and when slots are exchanged in the compression algorithm. We observe that all slot exchanges are instigated by a slot that is made available through a cancelled or a delayed flight. Such a slot leads to a series of slot exchanges, in which flights are repeatedly moved up in a way that maximizes the return for the releasing airline. To capture this type of exchange by a collection of offers, we introduce the following two types of offers.

- The *type 1* offer may be viewed as a default offer, and is depicted in Figure 10. This offer simply states that an airline would be willing to give up the slot

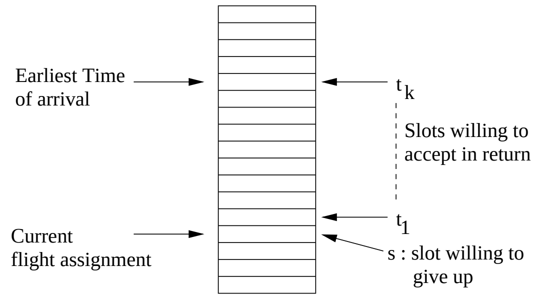

Figure 10: "Default" Offers

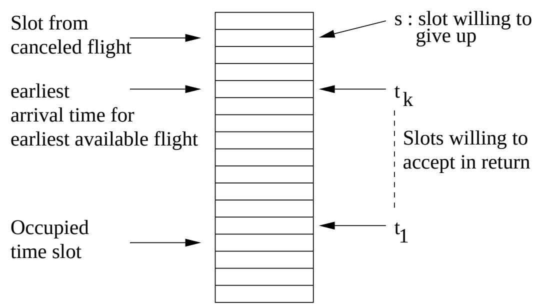

Figure 11: Offer associated with canceled or delayed flights

currently occupied by a flight, in return for an earlier slot, as long as the new slot is not earlier than the earliest time of arrival for the flight.

- The *type 2* offer applies when a flight is cancelled or delayed. In this case, the corresponding offer will be as depicted in Figure 11. Here, the releasing airline is willing to give up the slot in return for a reduction in the delay of a subsequent designated flight. As we will discuss, a single cancellation can lead to multiple offers of this type to effect a set of progressive moves for a single airline's flights.

In fact, an airline need not explicitly specify either of these offers. The first type is easily derived by observing the earliest time of arrival of each flight. The second type can be derived by combining the slot opened by a cancelled or delayed flight (offered slot) together with the slots that provide delay reduction for the nearest flight from the same airline (slots the airline is willing to accept in return).

Let us now examine the structure of the overall set of offers a single airline could generate. As before, we assume the flights are given by $\mathcal{F}$, and the slots by $\mathcal{S}$. For each non-cancelled flight f, we represent the earliest slot it can use as e_f, the slot it has currently been assigned to as i_f. In general, there will be one or more flight cancellations, which lead to un-occupied slots. To ease comparison with our previous model, we again characterize this situation using goal slots. Thus we further assume that a goal slot $g(f)$ is associated with each flight f.

Given this information, an airline would propose an offer for each non-cancelled flight f. Informally, the offer states that the airline is willing to exchange $g(f)$ for a reduction in the delay assigned to f. More specifically, each airline would be willing to a trade slot $g(f)$ for a slot in the range $e_f, \ldots, i_f - 1$. However, it may be necessary to insure against the possibility that the trade cannot be executed; in this case, the airline would like to maintain possession of the initial slot i_f to ensure flight f has a slot no later then its current slot. Thus, using the above mentioned framework we may associate with each non-cancelled flight f an offer

$$(g(f), \{e_f, \ldots, i_f - 1\}, \rho(g(f)) = i_f),$$

We note that this process leads to three kinds of offers.

- If $g(f) = i_f, \Longrightarrow$ type 1 offer.
- If $g(f) \geq e_f, \Longrightarrow$ type 2 offer.
- If $e_f < g(f) < i_f, \Longrightarrow$ combined type 1 and type 2 offers.

It should now be easy to see that with these offers, the constraint set from the assignment model in the previous section will be identical to the constraint set defined for the compression optimization model in Section 3.4. We note that in the bartering model case the items to be assigned are goal slots and in the compression optimization model they are flights. However, in each case there is one node per flight and the adjacent arcs are exactly the same.

The broker's (FAA's) problem can now be formulated and solved as an assignment problem as described in the previous section. The one remaining open question is determining the cost or value function associated with each exchange offer. We first note that any exchanges will be driven by type two offers, which involve giving up an earlier slot for a later one. Such offers will be "rare" compared with type 1 offers, which will be generated by default for any flight that has received a delay. Thus, the bartering mechanism should provide strong encouragement for type 2 offers. To do so a bilevel programming approach will be used in which an objective function associated with type 2 offers will be given priority of an objective function associated with type 1 offers. This can be achieved by multiplying the type 2 objective function by a large enough weight and then adding the two objective functions together.

In formulating the type 2 offer objective function, we start with a cost function that increases with the distance between the slot offered and the slot accepted in return. We then argue that this cost function should have a marginally increasing rate of increase; this gives priority to reducing longer deviations between the slot given up over the slot received in return over smaller deviations. This is a fairly standard notion of fairness and as we shall see is consistent with the earlier ideas presented in this paper.

With each offer of the first type (the default offers) we associate a value function that increases with delay reduction (the distance between the slot offered and the slot accepted in return). In this case we give the value function a marginally decreasing rate of value increase. Informally, the marginally decreasing rate of increase represents the objective of distributing the delay reductions enabled by a cancelled or delayed flight evenly among other flights. The combined objective function is to minimize the type 2 objective function times a large weight minus the type 1 objective function.

Let us now consider the cost function principles presented above and compare them to the cost function defined in Section 3.4. Consider an offer $(g(f), \{e_f, \ldots, i_f - 1\}, \rho(g(f))$, and let $t \in \{e_f, \ldots, i_f - 1\}$ (Here we assume that the combined type 1 and type 2 offers have been split into two separate purposes). The case where $t < g(f)$ corresponds to a type 1 offer, and should have a marginally decreasing value function, which could be

$$c_{g(f),t} = (s - g(f))^{1+\epsilon},$$

with $0 < \epsilon \ll 1$. On the other hand, the case where $t > g(f)$ corresponds to a type 2 offer which should have a marginally increasing cost function with higher priority. One possibility to achieve such as cost function would be to let

$$c_{g(f),t} = C(s - g(f)) + (s - g(f))^{1+\epsilon},$$

with $C \gg 1$. The resulting objective function therefore corresponds to the objective function we derived for the compression optimization models.

To summarize, this shows that the compression procedure has a natural interpretation as a form of mediated bartering, in which the FAA - acting as a mediator - matches offers by prioritizing offers to move down, thus implicitly rewarding airlines for offering to reduce congestion.

4.3 Extensions

The interpretation of compression as bartering and the associated optimization model invites several possible extensions to the current inter-airline slot exchange procedure.

Conditional Exchanges. In the current slot exchange framework, when airlines cancel flights they generally do not know in advance the benefits compression achieves for their remaining flights. This might limit their ability to make the best decision as to whether or not to cancel specific flights. Consider for example a small carrier, who has two flights in a GDP. He might prefer to cancel the first flight, but only if his second flight can leave on time. By allowing *conditional* cancellations, this decision could expressed as an exchange offer. More specifically, and airline could propose an exchange offer $\langle s; t_1, \ldots, t_k \rangle$ as usual, with s the proposed cancellation and $t_1, \ldots, t_k$ the range of slots required in return for canceling the flight associated with slot s. Now, if a slot in $t_1, \ldots, t_k$ were not available, the exchange will not be implemented and the airline will not cancel. Such conditional offers are easily included in the assignment model by setting $\rho(f) = f$.

"Continuous" Compression. A possible extension of conditional exchanges could be to run the inter- airline slot exchange as a continuous process. The process could start off with the default offers available. Subsequently, airlines could propose and submit offers at any time. The slot-exchange procedure could continuously search for and implement any acceptable set of exchanges. This could be

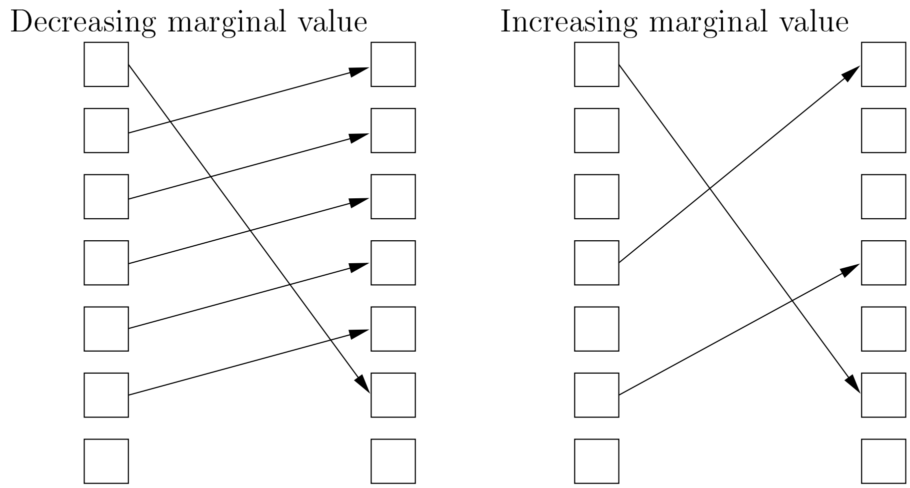

Figure 12: Effects of alternate value functions

extended with a query-capability that would enable airlines to evaluate alternate offers (cancellations) before actually submitting one.

Alternate Value Functions. In the compression algorithm, delay reduction are distributed over many flights, resulting in many, but small flight movements. In our model, this was captured by using a cost function with a decreasing marginal value (e.g. see Figure 12). One might however wonder whether this provides the "best" distribution of delay among flight. In many cases, the delay cost associated with a flight approximates a stepwise function, and therefore it might be argued that it would be preferable to distribute the delay reduction over fewer flights. Using the model we described before this could be achieved by using marginally *increasing* value function, resulting in schedules as shown on the right side of Figure 12.

Combined Exchanges. Another extension that could be beneficial to the airlines would be to allow more complex slot exchange offers, that is, offers that in which multiple slots are offered and multiple slots are required in return. To illustrate this, let us consider the example shown in Figure 13. Here, airline B's first flight is critical, while its second flight is less important. Airline A desires to move its two flights closer together (a common desire for flights within a bank). Consequently, combining two one-for-one exchanges as shown in Figure 13 is beneficial to both airlines. So, combined exchanges offer the possibility for more complex trade-offs on the airline side, which are not possible by basic one-for-one exchange offers. In this situation, the mediator's problem becomes more complex so that an assignment model is no longer adequate.

Side Payments. Probably the most dramatic change suggested by the bartering model involves the introduction of side payments. It would be natural to consider a side-payment when one airline relinquishes an early slot for a much later one. This case moves the analysis from a bartering arena to auctions. It would be

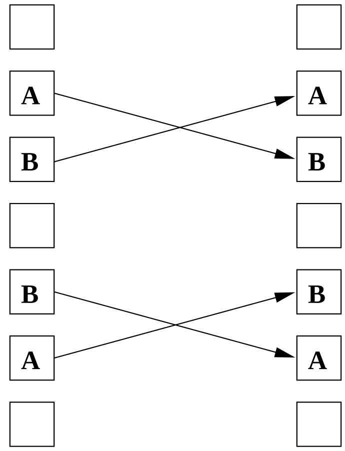

Figure 13: An example of combined exchanges

natural to then consider "pure" slot buying and selling, in addition to exchanges with side payments. Since each airline could potentially be both a buyer and seller, a double-auction mechanism would be appropriate. Clearly, there is a large set of both technical and political issues associated with this extension.

5 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.