Optimization and Mediated Bartering Models for Ground Delay Programs

Thomas Vossen[1] and Michael Ball[2]

[1]*Leeds School of Business, University of Colorado, Boulder, Colorado 80309*

[2]*Robert H. Smith School of Business and Institute for Systems Research, University of Maryland, College Park, Maryland 20742*

Received 9 February 2005; revised 24 August 2005; accepted 9 September 2005
DOI 10.1002/nav.20123
Published online 5 December 2005 in Wiley InterScience (www.interscience.wiley.com).

Abstract: The Federal Aviation Administration (FAA) and the airline community within the United States have 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 increase airline control over 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 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 possibilities for increased decision-making capabilities by the airlines. © 2005 Wiley Periodicals, Inc. Naval Research Logistics 53: 75–90, 2006.

Keywords: air traffic flow management; ground delay programs; collaborative decision making; equity

1. INTRODUCTION

The Federal Aviation Administration (FAA) and the major airlines in the United States have embraced a new paradigm for 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 [4, 23]. 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 applies to a wide range of applications in Air Traffic Flow Management, the primary focus so far has been the implementation of Ground Delay Program (GDP) enhancements.

The FAA issues GDPs whenever it expects sustained periods of congestion at an airport. Usually, this is caused by a sudden decline in airport arrival capacity due to adverse weather conditions. In a GDP, flights that are scheduled to arrive during the expected periods of congestion are assigned delays prior to their departure; hence, the term ground delay. The reason 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. The allocation of ground delays can be viewed as a resource allocation problem, in which available arrival capacity (i.e., a sequence of arrival slots) is allocated to flights. This problem is also known as the Ground Holding Problem and has received considerable attention. The basic version of the problem, in which both demand and airport capacity are assumed to be deterministic, was first studied by Odoni [16]. More recent models have, among others, addressed stochastic arrival capacity [5, 17], the incorporation of banking constraints [10], and the extension to en-route capacities [8]. 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 impact on individual airlines.

As a result, however, it is difficult to apply these models under the CDM paradigm (see also [1]). It is therefore hardly surprising that the GDP enhancements implemented under CDM follow quite a different approach (see [7] for background on the operation of GDPs under CDM). This approach is based on the consensus recognition that airlines have claims on the available arrival capacity, based on their

Correspondence to: M. Ball (mball@umd.edu)

original flight schedules. This realization has led to a fundamental change in the way arrival capacity is allocated, as well as 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). This procedure has removed disincentives airlines previously had to provide accurate information about delays and cancelations. 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 cancelations and attempts to do so in a fair and equitable manner. While the RBS and Compression procedures have an intuitive appeal, the underlying fairness concepts are largely left implicit. This complicates the incorporation of more advanced trade-offs, such as those proposed in various extensions of the Ground Holding Problem.

Our main objective in this paper is therefore to interpret and formalize 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. We describe how both the initial allocation of slots and the inter-airline slot exchange may be interpreted as an assignment problem with 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 a basis for expanding the set of trade-offs available to the airlines. By demonstrating that the criterion of lexicographically minimizing the maximum delay applies to both the ration-by-schedule algorithm and the Compression algorithm, we show that this criterion embodies a fundamental notion of fairness that has emerged from a variety of FAA–airline gaming exercises and negotiations over the past few years. In addition, we discuss an alternative interpretation of the process, in which the inter-airline slot exchange corresponds to a mediated bartering process. We compare the problem of matching the offers submitted by airlines with the current procedure for exchanging slots and discuss its potential benefits.

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.

2. GDP PROCEDURES

The FAA continuously monitors airports throughout the United States for capacity-demand imbalances. Whenever it predicts that the number of flights arriving at an airport within a 15-minute interval exceeds the capacity of the current runway configuration, FAA directives mandate a response [9]. 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 number of decisions, 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) can be derived, since en-route travel times are 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. This is due to flight delays and cancelations that may occur after the initial assignment of flights to slots. Flight delays are caused by mechanical and other operational problems. Flight cancelations, on the other hand, are mainly the result of airline internal schedule adjustments that aim to mitigate the adverse effects of a GDP on its operations. As such, a GDP is an iterative process as depicted in Fig. 1, in which the FAA repeatedly assigns flights to slots, based on airlines' flight updates, substitutions, and cancelations.

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. Although intuitively appealing, this method of assigning flights to slots penalized the airlines for providing accurate information. This effect, known as the double penalty issue, is best explained by the following example. Suppose a flight was originally scheduled to arrive at 10:00,

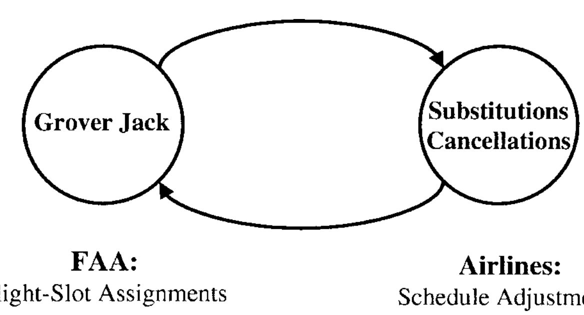

Figure 1. FAA/airline interaction in a GDP prior to CDM.

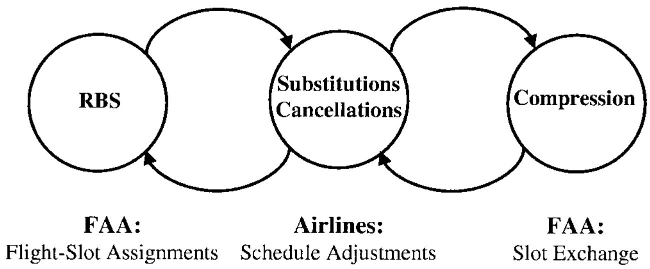

Figure 2. FAA/airline interaction in a GDP under CDM.

but experienced a delay of 30 minutes due to mechanical problems. If a GDP was implemented in which the flight was delayed for another 30 minutes, its total delay would therefore 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 result, airlines were hesitant to provide the FAA with accurate estimates of their flight delays. In addition, airlines had no incentives to report flight cancelations in a timely manner. As a result, GDP decisions were based on poor data, which led to inefficient programs.

The GDP enhancements implemented under CDM address these issues with a fundamental change in the allocation of capacity by the FAA. Instead of an assignment of flights to slots, the CDM "philosophy" considers the allocation of capacity to be an assignment of slots to airlines. This has led to the introduction of two new allocation mechanisms, called Ration-By-Schedule and Compression. The RBS algorithm generates an initial allocation of slots to airlines, based on the recognition that airlines have claims on the 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 cancelations and delays. Under CDM, the interaction between the FAA and the airlines can therefore be depicted as in Fig. 2. Initially, the FAA rations the arrival slots among the airlines. Subsequently, airlines can adjust their schedules by substituting and canceling flights. To ensure utilization, the FAA may periodically execute Compression. We also note that an airline's schedule adjustments may be followed by an execution of the RBS algorithm; these revisions may be necessary when the arrival capacity available changes (i.e., due to prolonged bad weather conditions, etc.).

In the remainder of this section we describe the RBS and Compression algorithms in greater detail, using the following notation:

- $F = \{f_1, \ldots, f_n\}$, the flights affected by the GDP;
- $S = \{s_1, \ldots, s_n\}$, the sequence of arrival slots in the GDP. For each slot s_j, t_j represents the slot time;
- A, the airlines involved in the GDP;
- $O : F \to A$, a mapping that defines the flight–airline ownership relation. For each airline $a \in A$, F_a represents the flights from that airline, i.e., $F_a = \{f \in F \mid O(f) = a\}$;
- oag_i for all $f_i \in F$, the originally scheduled arrival time of flight f_i;
- $erta_i$ for all $f_i \in F$, the (most recent) estimated arrival time of flight f_i;
- cta_i for all $f_i \in F$, the controlled time of arrival for flight f_i; that is, the flight's arrival time after the GDP has been implemented;
- $I : F \to S$, a mapping that defines the current assignment of flights to slots;
- $C \subseteq F$, the flights that have been canceled.

2.1. Ration-by-Schedule

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. The RBS algorithm, however, orders flights according to their *scheduled time of arrival* (as opposed to the *most recent estimated time of arrival* ordering used in Grover Jack). As a result, airlines do not forfeit a slot by reporting a delay or a cancelation, which was the case in the Grover Jack algorithm. The RBS algorithm can be outlined as shown in Fig. 3. The actual RBS algorithm must 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 [7, 11] for a discussion of these details).

Note that the resulting flight schedule may be inefficient in its utilization of arrival capacity. Arrival slots may have been assigned to flights that have been canceled or delayed and therefore cannot use their assigned slot. The end result of RBS should therefore not be viewed as an assignment of slots to flights, but rather as an assignment of slots to airlines. Airlines can reassign its flight-slot assignment using the cancelation and substitution process. It should be emphasized that this notion of slot ownership is one of the main tenets of the

---

RBS ALGORITHM:

Step 1. Order the flights in F by increasing scheduled time of arrival.
Go to step 2.

Step 2. Select the first flight in F that has not been assigned a slot.
If no such flight exists, the algorithm is terminated.
Otherwise, the flight is assigned the earliest unassigned slot it can meet.

Figure 3. The Ration-by-Schedule procedure.

CDM paradigm: there is a general consensus among airlines that this is indeed a fair method of rationing arrival capacity.

Finally, it is interesting to note that RBS can also be interpreted as a procedure that prioritizes flights according to their accrued delay (that is, a flight's expected arrival time minus its scheduled arrival time). This criterion has in fact been proposed [12] for the allocation of en-route traffic resources.

2.2. Compression

After a round of substitutions and cancelations, the utilization of slots can often be improved. Flight cancelations and delays may create "holes" in the current schedule; that is, there will be arrival slots that have no flights assigned to them. The Compression algorithm moves flights up in the schedule to fill these slots.

The idea behind the Compression algorithm is to reward airlines for slots they release, thus encouraging airlines to report cancelations. The extent to which flights can be moved up is limited, since a flight cannot depart before its scheduled departure time. The procedure also assumes that flights cannot be moved down from their position in the current schedule I. The set of slots $\{e_i, \ldots, I(f_i)\}$ therefore specifies the window in which flight f_i can land. A conceptual overview of the Compression Algorithm is shown in Fig. 4.

COMPRESSION ALGORITHM:

Step 1. Order the flights according to the current schedule. Determine the set of open slots C_S. For each slot $c \in C_S$, execute step 2.

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

- **2.1.** Determine the first flight f_j from airline a (in the current schedule) that can be assigned to slot c, that is, for which $c \in \{e_j, \ldots, I(f_j)\}$. If there is no such flight, go to Step 2.2. Otherwise, go to Step 3.
- **2.2.** Determine the first flight f_j from any other airline that can be assigned to slot c. If there is no such flight, go to Step 2.3. Otherwise, go to Step 3.
- **2.3.** There is no flight that can be assigned to slot c. Return to Step 1 and select the next open slot.

Step 3. Swap the slot assignments of flights f_i and f_j, i.e., assign flight f_j to slot c, and flight f_i to slot $I(f_j)$. Note that airline a now owns open slot $I(f_j)$. Next, slot $I(f_j)$ is made the current slot, and Step 2 is repeated.

Figure 4. The Compression procedure.

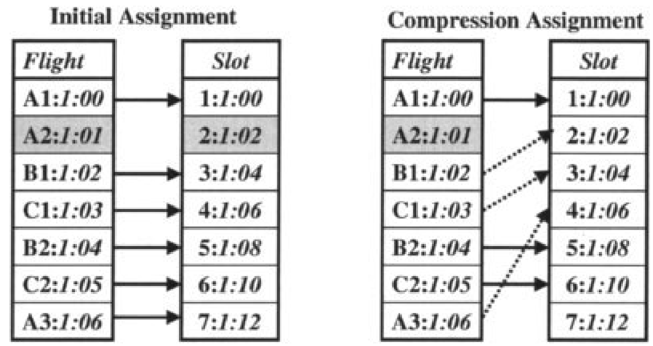

Figure 5. Compression example.

There are two ways for an arrival slot to become open; either the flight assigned to that slot has been canceled or it has been delayed by a cause unrelated to the GDP, e.g., mechanical delay. In either event, the controlling airline will release the slot to the Compression algorithm. 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 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 Fig. 5. 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 ($A2$). The rightmost figure shows the flight schedule after execution of the Compression algorithm: as a first step, the algorithm attempts to fill slot 2 owned by airline A. Since there is no flight from airline A that can use the slot, the slot is allocated to flight $B1$. This process is repeated with the next open slot, which is assigned to flight $C1$. Slot 4, however, can be used by airline A and will therefore be assigned to flight $A2$.

3. OPTIMIZATION MODELS FOR SLOT ALLOCATION AND EXCHANGE

The notion of fairness plays an important role in both the slot allocation and the exchange procedures introduced under CDM. 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 between airline and FAA traffic flow managers. As such, they embody consensus concepts for fair allocation within this setting. Unfortunately, however, the embodiment of fairness under CDM is largely left implicit in these procedures, and in fact, different and even conflicting concepts are sometimes used to describe these procedures. This not only generates confusion and complaints, but also complicates the introduction of CDM in more complex settings. 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 in the U.S. House of Representatives. 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, there are certain common principles that return in many real-world problems and that provide a constructive basis for reasoning about equity (see [22] for a comprehensive overview). Principles of equity are commonly based on pairwise comparisons, that is, allocations are evaluated by comparing pairs of claimants according to 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, such that the less deserving claims remains so after the transfer. The definition depends on the specification of who is less or more deserving. This can be formalized with a 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 used in a variety of resource allocation problems where it is desirable to allocate limited resources equitably among competing activities [14, 15]. 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, not be sufficient as they leave open many possibilities for allocating resources from among the activities that are not among the worst off. To address this issue, a lexicographic minimax objective function can be used. A lexicographic minimax solution allocates resources in such a way that no performance function can be improved by changing the performance of an activity that is already equal or larger. In this sense, the lexicographic minimax criterion corresponds to the absence of justified transfers.

In the remainder of this section, we aim to formalize the fairness concepts used under CDM. Fairness is achieved by formulating both the allocation and the exchange of slots as optimization problems that use the lexicographic minimax objective. We show that these models are equivalent or nearly equivalent to both RBS and Compression.

3.1. OPTIFLOW Model

The OPTIFLOW model [2] is an optimization model that assigns flights to slots so as to minimize overall delay costs. Therefore, the model follows a central planning paradigm and does not apply under CDM. Nevertheless, the model provides the starting point for the optimization models described later in this section. The OPTIFLOW model is formulated as an assignment problem, with

- decision variables $x_{ij} \in \{0, 1\}$, where $x_{ij} = 1$ if flight f_i is assigned to slot s_j and $x_{ij} = 0$ otherwise;
- cost coefficients c_{ij} that represent the "cost" of assigning flight f_i to slot s_j,

for all $f_i \in F/C, s_j \in S$ such that $t_j \geq erta_i$. The Linear Programming formulation of the problem is as follows.

$$\text{Min} \quad \sum_{f_i \in F/C, s_j \in S: t_j \geq erta_i} c_{ij} x_{ij}$$

subject to:

$$\sum_{s_j \in S: t_j \geq erta_i} x_{ij} = 1 \qquad \text{for all } f_i \in F/C$$

$$\sum_{f_i \in F/C} x_{ij} \leq 1 \qquad \text{for all } s_j \in S$$

$$x_{ij} \in \{0, 1\}$$

The constraints express that each flight that is not canceled is assigned to a slot and that each slot is assigned to at most one flight. The cost coefficients are expressed as

$$c_{ij} = w_i (t_j - oag_i)^{1+\epsilon},$$

with w_i a weight associated with flight i and $0 < \epsilon < 1$. Parameter ϵ results in super-linear growth of the tardiness cost of a flight, so that the model favors assigning moderate amounts of delay to two flights rather than assigning a large amount of delay to one flight and a small amount to another. Suppose, for example, 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. The weights w_f in the cost coefficients represent the cost associated with the ground and airborne delay for that flight. 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 $f \in F$, the OPTIFLOW model will find a solution that minimizes the total delay

$$\sum_{f_i \in F/C, s_j \in S: t_j \geq erta_i} (t_j - oag_i)x_{ij}.$$

Moreover, the slots that have flights assigned to them are identical for any delay-minimizing solution.

PROOF: First, we note that the total delay of assignment only depends on the slots used, since the scheduled arrival times are constants. Now let x^* be an optimal solution to the OPTIFLOW model, but suppose that it not does not minimize overall delay. Let y^* be any optimal delay minimizing solution and consider the symmetric difference $x^* \oplus y^* = \{(i,j) : x^*_{ij} + y^*_{ij} = 1\}$ of the arcs in x^* and y^*, that is, the set of arcs (flight-slot assignments) belonging to either x^* or y^* but not both. The graph induced by $x^* \oplus y^*$ consists of a collection of even alternating cycles and 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^*. Thus, there must be at least one even alternating path:

$$s_{j(0)} - f_{i(1)} - s_{j(1)} - f_{i(2)} \cdots f_{i(K)} - s_{j(K)}$$

with $(f_{i(k)}, s_{j(k-1)}) \in y^*$ and $(f_{i(k)}, s_{j(k)}) \in x^*$. By construction slot $s_{j(0)}$ is uncovered in x^* and so by the optimality of x^* we have that $s_{j(k)} < s_{j(0)}$ for all $1 \leq k \leq K$, since otherwise x^* could be improved by changing a flight assignment $(f_{i(k)}, s_{j(k)})$ to $(f_{i(k)}, s_{j(0)})$. To see this, we observe first that $s_{j(1)} < s_{j(0)}$ (otherwise x^* could be improved by allocating $f_{i(1)}$ to $s_{j(0)}$). Now consider $s_{j(2)}$. If $s_{j(2)} > s_{j(0)}$, x^* could be improved by assigning $f_{i(2)}$ to $s_{j(0)}$ (note that this assignment is feasible since $s_{j(1)} < s_{j(0)}$ and $f_{i(2)}$ is assigned to $s_{j(1)}$ in y^*). Thus, we also have $s_{j(2)} < s_{j(0)}$. Repeating this argument yields the result for all $1 \leq i \leq k$. However, the fact that $s_{j(k)} < s_{j(0)}$ implies that y^* is not an optimal solution, which contradicts our assumption. Thus, x^* is a delay-minimizing solution.

A similar argument shows that any delay-minimizing solution assigns flights to the same slots. 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 $s_{j(0)}$ used in x^* and one end-node $s_{j(k)}$ used in y^*. This, however, implies that one of the solutions does not 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 leaves room to consider other criteria, e.g. equity, in addition to delay minimization.

3.2. Optimization-Based Slot Allocation

In this section, we show that both the RBS algorithm and a special case of the OPTIFLOW model result in flight-slot assignments that lexicographically minimize the maximum flight delays. To state this equivalence, we let T represent the maximum delay allocated by the RBS algorithm and define for each $t = 0, 1, \ldots T$ the performance function

$$d_t = |\{f_i \in F : cta_i - oag_i = t\}|.$$

THEOREM 3.2: The flight-slot assignments obtained by the RBS algorithm and by the OPTIFLOW model with $w_i = 1$ for all $f_i \in F$ lexicographically minimize the maximum delay with respect to the original flight schedule; that is, both approaches yield an allocation that 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:

- *The assignment obtained by the RBS algorithm lexicographically minimizes the maximum delay.*

  To prove this, we let I_1 be a lexicographical min-max assignment and I_2 an assignment generated by the RBS algorithm and argue that both I_1 and I_2 assign the same flight to the first slot. Suppose this is not the case, i.e., we have $I_2(f_i) = s_1$ but $I_1(f_i) = s_k$ for some i and $k > 1$, and $I_1(f_j) = s_1$ with $j \neq i$. It follows from the RBS algorithm that $oag_i < oag_j$, and therefore $\max\{t_1 - oag_i, t_k - oag_j\} < \max\{t_1 - oag_j, t_k - oag_i\}$. This, however, contradicts the optimality of I_1, since the lexicographical min-max objective function can be improved by interchanging the assignment of f_i and f_j. To complete the proof, we can repeat this argument for slots $2, \ldots, n$.

- *The assignment obtained by the RBS algorithm is optimal w.r.t. the OPTIFLOW model.*

  This follows by a similar exchange argument. Let I_1 be an optimal solution to the OPTIFLOW model and I_2 an assignment generated by the RBS algorithm. Suppose again this is not the case, i.e., we have $I_2(f_i) = s_1$ but $I_1(f_i) = s_k$ for some i and $k > 1$, and $I_1(f_j) = s_1$ with $j \neq i$. This, however, would imply that interchanging the assignment of f_i and f_j would improve the OPTIFLOW objective function (since $oag_i < oag_j$), which is a contradiction. Again, repetition of the argument yields the desired result. □

In other words, the allocation obtained by both 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 a candidate upon which to base the allocation of resources in other contexts. In particular, using a lexicographical minimax objective might prove useful if the allocation involves more complex combinations of resources (e.g., a region of airspace), which require the solution of a more complex optimization problem.

3.3. Optimization-Based Slot Exchange

The inter-airline exchange of slots in GDPs is performed by the Compression algorithm, which aims to reward airlines for slots they release. In this section, we first represent the inter-airline exchange of slots as an assignment problem with an appropriately defined lexicographic minimax objective. Subsequently, we show that this problem can be solved with a greedy algorithm that closely resembles the Compression algorithm.

To specify a lexicographic minimax objective that captures the exchange of slots in Compression, it is necessary to define a set of performance functions together with appropriate targets or goals. To incorporate the notion that airlines are rewarded for slots they release, we associate with each (non-canceled) flight a *goal* slot. The performance functions measure 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:

- $g : F/C \rightarrow S$, such that $g(f_i) \in I(F_a)$ when $O(f_i) = a$,

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

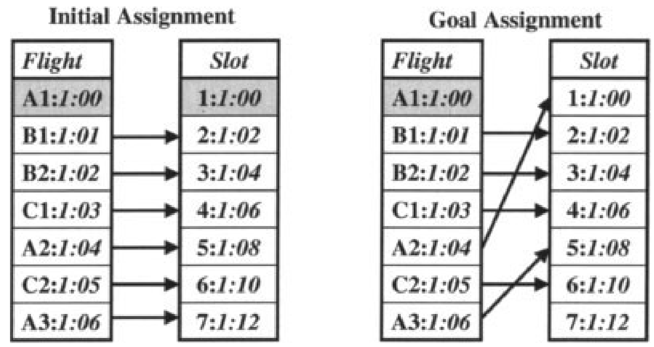

Figure 6. Example of initial allocation and assignment of goals.

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 illustrate this concept, let us consider the example depicted in Fig. 6. Again, each flight has an associated earliest time of arrival, and each slot has an associated slot time. In the initial assignment depicted in Fig. 6, airline A has cancelled its first flight, leading to an open slot. Ideally, it would assign its next flight, $A2$, to this slot. Thus, slot 1 represents the goal for flight $A2$ (note that $A2$'s earliest arrival time is later than 1:00). Subsequently, A would like to assign $A3$ to slot 5, which is now vacated by $A2$. Thus, slot 5 represents the goal for flight $A3$.

In general, there may be different 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 times of arrival and the goals according to increasing slot times. Then, we assign the first flight to the first goal, second flight to the second goal, etc. In our experiments, 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.

Given a flight-goal assignment, represented by g, the next step is to determine an objective function for the OPTIFLOW model that lexicographically minimizes each flight's deviation from its goal slot. For this, we define the performance functions

$$D_k(x) = \sum_{f_i \in F, s_j \in S: t_j - t_{g(f_i)} = k} x_{ij}$$

for $k_{\min} \leq k \leq k_{\max}$, where $k_{\max}$ represents the maximum (positive) goal deviation and $k_{\min}$ the minimum (and potentially negative) 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 such that higher goal deviations have higher priority. As such, our objective is to lexicographically minimize the vector

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

over all possible flight-slot assignments x. This can be achieved by using an objective function that is an appropriate weighted sum of the performance functions [18], that is,

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

The weights W_k must ensure that whenever a solution x_1 is lexicographically larger than a solution x_2 (according to vector (1)), its costs will also be higher. In our case, an appropriate choice of weights is

$$W_k = (k - k_{\min})^{(1+\epsilon)}$$

for $k_{\min} \leq k \leq k_{\max}$ and $\epsilon > 0$. Observe that W_k is strictly increasing, since $k \geq k_{\min}$, and strictly convex in k. As a result, the overall objective function coefficients in the OPTIFLOW model will be

$$c_{ij} = (t_j - t_{g(f_i)} - k_{\min})^{(1+\epsilon)}, \tag{2}$$

for $f_i \in F$ and $s_j \in S$. The following theorem then ensures the desired result.

THEOREM 3.3: The OPTIFLOW model using objective function coefficients (2) yields a solution that lexicographically minimizes (1). Moreover, using objective function coefficients (2) the solution obtained by the model also minimizes total delay.

PROOF: The proof is similar to Theorem 3.2. That is, we let I_1 be an assignment that lexicographically minimizes (1) and I_2 an optimal assignment generated by the OPTIFLOW model using objective function coefficients (2). As before, we argue that both I_1 and I_2 assign the same flight to the first slot. Suppose this is not the case, i.e., we have $I_1(f_i) = s_1$ but $I_2(f_i) = s_k$ for some i and $k > 1$, and $I_2(f_j) = s_1$ with $j \neq i$. Since I_1 lexicographically minimizes (1), it follows that $g(f_i) < g(f_j)$ (note that no two flights will have the same goal). Since W_k is stricly increasing and strictly convex, we therefore have

$$c_{i1} - c_{j1} < c_{ik} - c_{jk}$$

or

$$c_{i1} + c_{jk} < c_{ik} + c_{j1}.$$

This, however, contradicts the optimality of I_2, since the objective function value of the OPTIFLOW model can be improved by interchanging the assignment of f_i and f_j. To complete the proof, we again repeat this argument for slots $2, \ldots, n$. Finally, minimization of total delay follows from Theorem 3.1. □

The resulting OPTIFLOW model corresponds to an assignment problem and can therefore be solved efficiently. However, a simpler procedure that finds optimal solutions exists and is shown in Fig. 7. This procedure repeatedly assigns the next available slot to the flight that has the earliest remaining goal slot among all flights that can use the slot. The correctness of the procedure is insured by the following theorem.

Init :
  Let $x_{ij} := 0$ for all $f_i \in F/C, s_j \in S$;
  Let $F' := F/C$

For $j \in 0, \ldots, n-1$ **Do**
  Let $i' := \arg\min_{f_i \in F': erta_i \leq t_j} g(f_i)$;
  { $f_{i'}$ is the flight with the earliest goal among those that can use slot s_j }
  Let $x_{i'j} := 1$, $F' := F'/\{f_{i'}\}$

Od

Figure 7. Greedy algorithm for optimization-based slot exchange.

THEOREM 3.4: A solution x obtained by the greedy algorithm shown in Fig. 7 is an optimal solution for the OPTIFLOW model using objective function coefficients (2) and therefore yields a solution that lexicographically minimizes (1).

PROOF: This follows by an exchange argument, as in Theorem 3.2. Suppose x is an optimal assignment to the greedy algorithm, y an optimal solution to the OPTIFLOW model, and let s_j be the first slot at which they differ; that is, $x_{i_1 j} = 1$ and $y_{i_2 j} = 1$ with $i_1 \neq i_2$. Observe that, by construction of the greedy algorithm, it follows that $g(f_{i_1}) < g(f_{i_2})$. But, this would imply that we could reduce the cost in the OPTIFLOW model by interchanging the assignments of f_{i_1} and f_{i_2} in y, which contradicts the assumption that y is optimal. □

Another, more airline-based, interpretation of the procedure would be that each airline is assigned a set of (remaining) priorities corresponding to its goal slots. Then, the procedure repeatedly assigns the next available slot to the airline that has the highest remaining priority among all airlines that can use

Init :
Let $x_{ij} := 0$ for all $f_i \in F/C, s_j \in S$;
Let $l_i := t_{I(f_i)}$ for all $f_i \in F/C$;
Let $F' := F/C$

For $j \in 0, \ldots, n-1$ **Do**
Let $i' := \arg\min_{f_i \in F': erta_i \leq t_j} l_i$;

If $l_{i'} = t_j$ **Then**
Let $x_{i'j} := 1, F' := F'/\{f_{i'}\}$;

Else
Let $i' := \arg\min_{f_i \in F': erta_i \leq t_j} g(f_i)$;
{ $f_{i'}$ is the flight with the earliest goal among those that can use slot s_j }
Let $x_{i'j} := 1, F' := F'/\{f_{i'}\}$

Od

Figure 8. Adjusted Greedy algorithm for optimization-based slot exchange.

the slot. In either case, however, this result clearly indicates the difference between the Compression algorithm and the optimization-based slot exchange procedure. Whereas the Compression algorithm assigns flights to slots according to the order in which slots are vacated, the greedy procedure assigns flights in the chronological order of slots. As a result, however, the greedy procedure may violate a basic property of the Compression algorithm, that is, airlines can potentially lose slots they own and can use. This, however, can be prevented by including the constraints

$$x_{ij} = 0 \qquad \text{for all } f_i \in F/C, t_j > t_{I(f_i)}, \tag{3}$$

in the OPTIFLOW model, for a given initial assignment I. The resulting optimization problem can again be solved by a greedy procedure, which is shown in Fig. 8. The correctness of this procedure is insured by the following theorem.

THEOREM 3.5: A solution x obtained by the adjusted greedy algorithm shown in Fig. 8 is an optimal solution for the OPTIFLOW model using objective function coefficients (2), with the added constraints (3).

PROOF: The proof again follows using an interchange argument. Suppose x is not an optimal solution to the IP formulation. Then, there exists another solution y that is optimal and differs from x in at least one position. Let j_1 be the first position at which the allocations differ, and let i_1, i_2 be such that $x_{i_2,j_1} = 1$ and $y_{i_1,j_1} = 1$. It follows that $l_{i_1} > t_{j_1}$ and $l_{i_2} > t_{j_1}$, that is, neither of the flights is due at time t_{j_1}. By construction we also know that $g(f_{i_2}) < g(f_{i_1})$, since the greedy algorithm will select the flight with the earliest goal among those that can use slot s_{j_1}. Moreover, $y_{i_2,j_2} = 1$ for some $j_2 > j_1$ such that $l_{i_2} \geq t_{j_2}$.

Suppose now that $l_{i_1} \geq t_{j_2}$. In that case, however, an interchange of the flights using the argument given in Theorem 3.4 (note that the interchange will not violate latest arrival times) would reduce the cost in the OPTIFLOW model and imply a contradiction.

Now consider the case where $l_{i_1} < t_{j_2}$. Thus, flight i_1 is due before time j_2, but after time j_1. Now let us look at all the flights $f_{i'}$ occupying the positions $j_1 + 1, \ldots, j_2 - 1$, and suppose all these flights were due before time j_2, e.g., $l_{i'} \leq j_2$. This implies that $|\{f \in F : I(f) = s_{j'}\}| > 1$ for at least one $j' \in j_1 + 1, \ldots, j_2 - 1$, since we know that $l_{i_1} > t_{j_1}$. This, however, contradicts our assumption that the initial assignment is feasible and therefore there is at least one $i' \in F, j' \in j_1 + 1, \ldots, j_2 - 1$ such that $y_{i',j'} = 1$ and $l_{i'} \geq j_2$. Again using the arguments from Theorem 3.4, we can interchange the assignments of f_{i_2} and $f_{i'}$ in y without increasing the cost. This yields a new assignment y where the distance between f_{i_1} and f_{i_2} has decreased. Thus, by repeating this argument we would eventually be able to interchange the flights such that f_{i_2} would be assigned to j_1, which shows that x is an optimal solution to the IP. □

3.3.1. Empirical Results

We now compare the performance of the optimization-based approach with the Compression algorithm. Our objective is to analyze how the difference in ordering the allocation of slots impacts the resulting slot allocations and the distribution of the resulting delays. To analyze these differences, we analyzed a number of historical GDPs at both Boston's Logan Airport and Chicago's O'Hare Airport during the first 4 months of 2001. For each of these GDPs, we considered the first execution of the Compression algorithm. For the current allocation of slots to airlines, we first assigned each airline's flights to its slots according to the flights' earliest arrival times. Slots that could not be assigned were left open. Given this initial allocation, we then compared the allocations that would have been obtained both by the Compression procedure and the adjusted Greedy procedure. The results are shown in Fig. 9, which depicts for each GDP both the total number of non-canceled flights and the flights that would have received different slots.

These results show that the optimization-based approach yields allocations that are very similar to the Compression procedure. For 57% of the instances the resulting allocations are in fact identical, while for 92% of the instances the allocations differ by at most 4.4%. Even for the three instances where a substantial number of flights receive different slots, the resulting allocations are close if we consider the differences in delay for these flights. For instance, at O'Hare airport on January 15, 2001, the procedures resulted in different slot allocations for 24.1% of the flights. Yet the average difference in delay for these flights was only 3.7 minutes, and for only

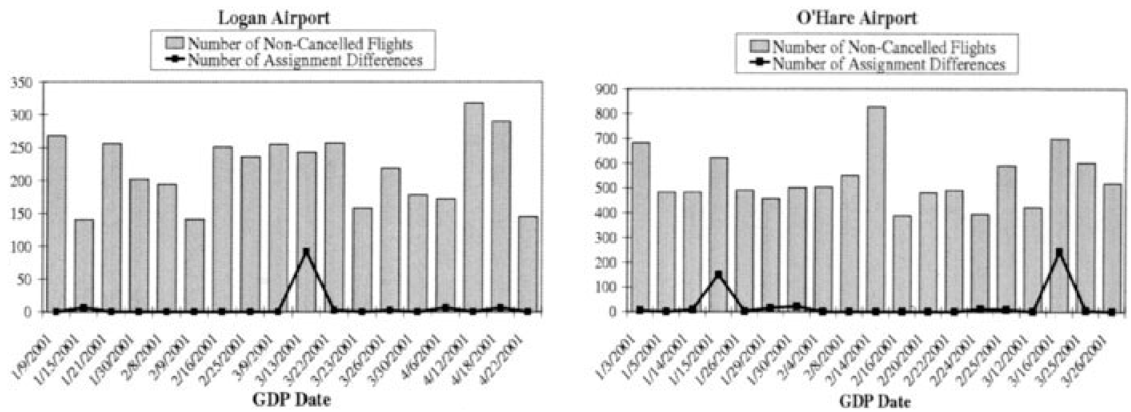

Figure 9. Compression vs. adjusted Greedy: Relative differences.

4% of these flights the delay difference was more than 10 minutes. At O'Hare airport on March 16, 2001, the average delay difference was 2.7 minutes, and for only 1.2% this difference more than 10 minutes. At Logan airport on March 13, 2001, the average delay difference was 7.3 minutes, and for 10.1% of the flights the difference was more than 10 minutes.

We also analyzed the performance of the original greedy procedure, that is, without the provision that no flight will involuntarily lose a slot it owns and can use. We found, however, that adding the constraints (3) had little impact. For 70% of the instances the results were identical to the adjusted greedy procedure, while for all but one instance at most six flights were affected.

3.4. Implications

Overall, these results indicate that appropriately defined lexicographic minimax objectives closely capture the fairness trade-offs used in both the initial allocation of slots to airlines and in the inter-airline exchange of slots currently employed by the FAA. Moreover, the greedy procedures used to solve the resulting optimization models highlight the underlying fairness principles. Each airline owns a given set of priorities (goals), which are assigned to its flights. The allocation procedures assign slots to flights according to these priorities.

In fact, one could argue that the greedy procedures provide a fundamental improvement, in that the Compression algorithm arbitrarily may or may not give an airline with an open slot certain priority relative to later slots by the way each Compression iteration cascades down. To illustrate this issue, let us consider the example in Fig. 10. Again, each flight has an associated earliest time of arrival, and each slot has an associated slot time. In the initial assignment depicted in Fig. 10, airlines A and B have canceled their first flight, leading to open slots. In the first iteration of the Compression algorithm, flight $D1$ is assigned to $S1$, flight $A2$ to $S4$, and flight $B2$ to $S5$. As a result, however, airline A will relinquish its priority over airline C in the second iteration. In this iteration, the Compression algorithm will assign flight $C1$ to $S2$, flight $A2$ to $S3$, and flight $B2$ to $S4$. Observe that airline C will receive an earlier slot than airline A in the final allocation. The Greedy procedure, on the other hand, respects priorities throughout and will therefore assign slot

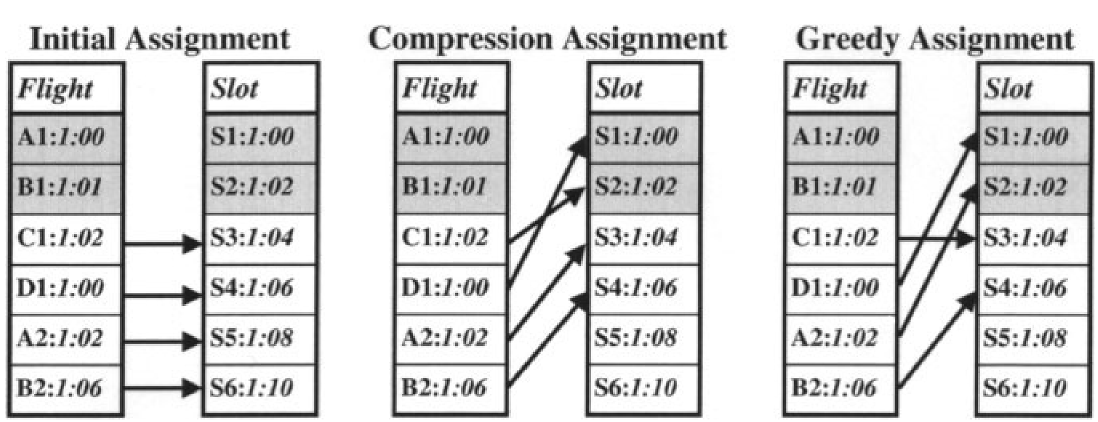

Figure 10. Example Compression vs. Greedy.

$S2$ to airline A. It seems clear that the solution produced by the Greedy procedure in Fig. 10 dominates the solution produced by Compression in that the Greedy solution gave a greater benefit in exchange for a canceled flight.

Viewing compression as a procedure that allocates slots according to airline-owned priorities provides a conceptual foundation that enables extension of the ideas to other settings. A simple immediate extension would be to provide the airlines with more flexible inputs to the process. That is, it would be possible for airlines to assign goals (priorities) to flights in different ways. For example, the airlines could specify a flight ordering criteria; the ordered list of priorities could then be matched to the ordered list of flights (to mimic Compression, we used a default ordering based on earliest time of arrival). Alternatively, the airlines could provide an explicit assignment of priorities to flights, thus providing an additional degree of control.

Even richer applications can be derived by interpreting the goals as the fundamental standard against which to measure any slot allocation. We have applied this idea to mitigate biases related to the manner in which certain flights are exempted from GDPs. Flight exemptions are employed due to the dynamic and stochastic nature of GDPs; generally speaking, flights originating from distant airports are often exempted to limit the risk of unnecessary delays (e.g., if weather conditions were to improve). As a result, however, the total delay assigned in a GDP will have to be distributed over fewer flights. This can have a significant impact, in that it introduces biases that systematically penalize certain airlines [19, 21]. Extensions to the procedures described in Section 3, however, can be used to mitigate these biases. Finally, we also note that, in addition to the definition of goals, it would also be possible to incorporate other fairness standards. Potential alternatives, as well as their impact on the resulting slot allocations, are discussed by Vossen [19].

4. TRADING MODELS FOR INTER-AIRLINE SLOT EXCHANGE

The GDP processes employed under CDM define the interaction between the airlines and the FAA. At the start of a GDP, 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 exchange is limited: given their preferences (which are provided by the earliest arrival times of each flight), Compression follows a simple procedure for (re)allocating slots.

As such, a relevant question is how the delay savings are distributed among flights in the exchange of slots and whether airline control could be extended so that a more beneficial exchange of slots could be achieved. Here, we address this issue by re-examining the role of the Compression algorithm. 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 that increase the airlines' flexibility 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 among 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, A, where each agent, $a \in A$, owns a set of goods, 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, $s \in S_a$, 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, E, is the set of possible exchange pairs, i.e.,

$$N = \bigcup_{a \in A} S_a \qquad E = \{(s,t) : s \in N \text{ and } t \in T_s\}.$$

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_1, a_2, and a_3 form a directed cycle. Then, an exchange is possible in which a_1 transfers good 1 to a_2, a_2 transfers good 2 to a_3, and a_3 transfers good 3 to a_1.

As a result, the problem of finding a feasible set of exchange sequences is equivalent to finding a set of non-intersecting directed cycles in (N, E). 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

$$\text{Min:} \quad \sum_{s \in N} \sum_{t \in T_s} c_{st} x_{st} \tag{4}$$

i due vincoli legati dicono: per ogni cessione c'è un acquisto

$$\text{s.t.} \quad \sum_{t \in T_s} x_{st} + y_s = 1 \qquad \text{for all } s \in N \tag{5}$$

$$\sum_{t : s \in T_t} x_{ts} + y_s = 1 \qquad \text{for all } s \in N \tag{6}$$

$$x_{st}, y_s \in \{0,1\} \qquad \text{for all } (s,t) \in E \text{ and } s \in N. \tag{7}$$

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 $x_{t_2 s} = 1$ 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 (6) and (7). 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 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 $\rho(s) \in S_a$ 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 new optimization model is then obtained by replacing constraint (6) with

$$\sum_{t : s \in T_t} x_{ts} + y_{\rho^-(s)} = 1 \qquad \text{for all } s \in N. \tag{8}$$

We note that 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 its offers. Without this requirement, it would be impossible to guarantee that the situation where all agents keep their goods, e.g., the solution where $y_{\rho^-(s)} = 1$ for all $s \in N$, would yield a feasible solution.

Figure 11 illustrates one cycle in a solution obtained under this model. The nodes represent goods, the solid arcs offers contained in E, and the dotted arcs depict the goods insured if offers are not accepted. The arcs in bold depict the cycle obtained by the 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:

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 Fig. 11, $\rho(s_2) = s_3$ 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

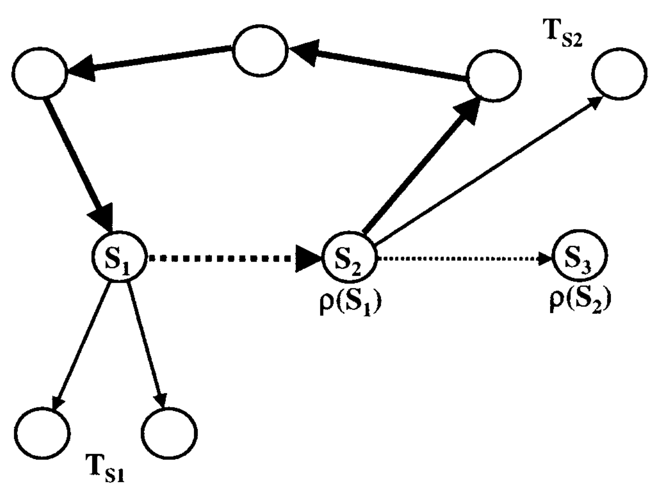

Figure 11. Cycle from solution to mediator's problem using "ρ-arc."

only "ρ"-arcs were chosen then this is equivalent to all of its offers being rejected. Clearly, a major component of this model is the definition of the objective function; generally speaking the definition may be application specific.

4.1.1. 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, A, is made up of the participating airlines; and 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 before. 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 canceled 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 Fig. 12. This offer simply states that an airline would be willing to give up the slot 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.

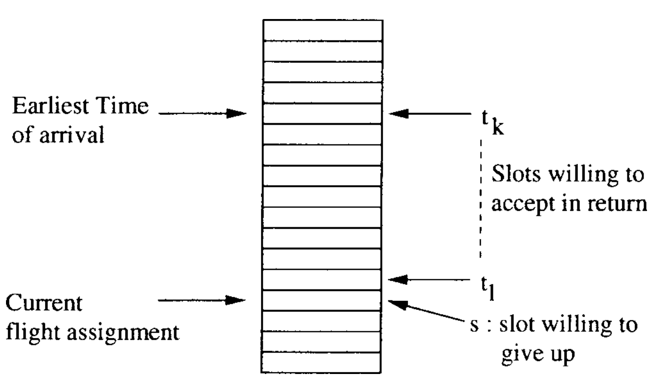

Figure 12. "Default" offers.

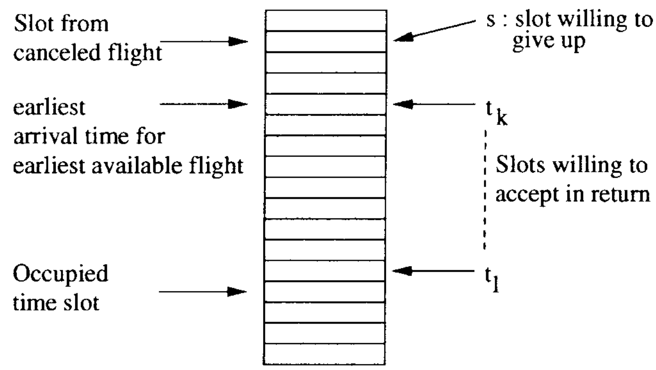

Figure 13. Offer associated with canceled or delayed flights.

- The *type 2* offer applies when a flight is canceled or delayed. In this case, the corresponding offer will be as depicted in Fig. 13. 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 cancelation 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 canceled 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 F, the slots by S, and for each non-canceled flight f_i, its earliest arrival time by $erta_i$ and its current slot by $I(f_i)$. In general, there will be one or more flight cancelations, 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 $g(f_i)$ is associated with each flight f_i. Given this information, an airline would propose an offer for each non-canceled flight f_i, which states that the airline is willing to exchange $g(f_i)$ for a reduction in the delay assigned to f_i. Thus, each airline would be willing to trade slot $g(f_i)$ for a slot in the range $\{s_j \in S : erta_i \leq t_j < t_{I(f_i)}\}$. 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 $I(f_i)$ to ensure flight f_i has a slot no later then its current slot. Thus, using the above-mentioned framework we may associate with each non-canceled flight f_i an offer

$$(g(f_i), \{s_j \in S : erta_i \leq t_j < t_{I(f_i)}\}, \rho(g(f_i)) = I(f_i)),$$

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

- If $t_{g(f_i)} = t_{I(f_i)}, \Longrightarrow$ type 1 offer.
- If $t_{g(f_i)} \leq erta_i, \Longrightarrow$ type 2 offer.
- If $erta_i < t_{g(f)} < t_{I(f_i)}, \Longrightarrow$ combined type 1 and type 2 offers.

It should now be easy to see that with these offers, the constraint set from the OPTIFLOW model with the added constraints (3) will be identical to the constraint set defined for the trading model. 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. In formulating the type 2 offer objective function, we therefore start with a cost function that increases with the distance between the slot offered and the slot accepted in return. We 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 canceled or delayed flight evenly among other flights. Let us now consider the cost function principles presented above and compare them to the cost function defined in Section 3.3. Consider an offer $(g(f_i), \{s_j \in S : erta_i \leq t_j < t_{I(f_i)}\}, \rho(g(f_i)))$ and consider a slot in $\{s_j \in S : erta_i \leq t_j < t_{I(f_i)}\}$ (here a single offer represents both a type 1 and type 2 offer). The case where $t_j < t_{g(f_i)}$ corresponds to a type 1 offer and should have a marginally decreasing value function, which holds for the objective function coefficients (2) defined in Section 3.3. On the other hand, the case where $t_j > t_{g(f)}$ corresponds to a type 2 offer, which should have a marginally increasing cost function with higher priority. Again, this holds for the objective function coefficients we defined before. 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.

4.2. Implications

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

Dynamic Trading

Compression and the bartering framework we have described here operate in a "batch" mode, where airlines submit cancelations and other information, which serves as input to a periodic execution of the mediator process (the compression algorithm). Thus, there could be a substantial lag between the time when information is input and results are received. A related issue is that, generally, when airlines cancel flights, they do not know in advance the benefits achieved for their remaining flights. This might limit their ability to make the best decision as to whether to cancel specific flights. Consider for example a small carrier, who has two flights in a GDP. This carrier might prefer to cancel the first flight, but only if the second flight can leave on time. By allowing *conditional* cancelations, this decision could be expressed as an exchange offer. More specifically, an airline could propose an exchange offer $\langle s; t_1, \ldots, t_k \rangle$ as usual, with s the proposed cancelation 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 $\rho(f) = f$. Alternatively, one could envision a setting where conditional "trades" are continuously proposed evaluated, and accepted or rejected. Such a dynamic trading environment would operate in a mode similar to a stock exchange. In fact, the FAA and airlines have recently adopted a *slot credit substitution (SCS)* process [6, 13], which effectively implements this idea.

Alternate Mediator Objective Functions

The objective function for the mediator model we have described closely represents the compression logic. One might question whether the current approach could be improved upon. Specifically, our model and compression typically produce small delay decreases for a large number of flights. This is the result of the marginally decreasing rate of value increase for type 1 offers. From an airline perspective, a decrease in flight delay by less than 10 minutes, in most cases, is of little value. This leads one to consider the merits of an objective function that produces larger delay decreases over a smaller number of flights. In fact, policies of this type are being considered within the new SCS process.

Complex Trades

The bartering framework we have described is based on "simple" offers where one slot is given up and another received in return. A very natural extension would be to allow more complex slot exchange offers, in which multiple slots are offered and multiple slots are required in return. For instance, one possibility could be that an airline would offer to delay one (non-critical) flight in return for a delay reduction on another (more important) flight. Note that such an offer would entail giving up two slots in return for two other slots. While allowing this type of offer would complicate the FAA's mediation problem, the trade-offs available to the airlines increase. In fact, such a scheme has been described in [19] and [20], where it is shown that the benefits to airlines could be quite substantial. These papers also provide computationally efficient integer programming models for the mediator's problem.

A True Exchange—Trading with Side Payments

Of course, any time goods are exchanged one would naturally expect monetary payments to be involved. In the slot exchange setting there are a number of issues to be addressed, such as a formal definition of property rights, before slot buying and selling on a day of operations could be implemented. Nonetheless, fundamental economic principles would suggest that substantial benefits could result. One might be tempted to replace the entire bartering operation with a marketplace in which slots are bought and sold. We would argue that, while simple buys and sells should be supported, it would be important to keep the basic slot-for-slot trading structure. In the most typical case an airline would not be interested in substantially changing the number of slots available to it; rather, the airline would be interested in paying for delay reduction or receiving compensation for delay increases. Thus, a market that supports true exchange transactions (slot A exchanged for slot B plus a side payment) is called for. While there are many challenges to instituting such a trading structure, we feel the benefits could be substantial. Reference [3] describes a basic framework for such a market and defines associated research questions.

ACKNOWLEDGMENTS

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

REFERENCES

[1] M. Adams, S. Kolitz, J. Milner, and A. Odoni, Evolutionary concepts for decentralized air traffic flow management, Air Traffic Contr Q 4 (1996), 281–306.
[2] M.O. Ball, R. Dahl, L. Stone, and T. Thompson, OPTIFLOW build-I design document, version 1.5, Optiflow Project Report to the FAA, December 1993.
[3] M.O. Ball, G. Donohue, and K. Hoffman, "Auctions for the safe, efficient and equitable allocation of airspace system resources," in P. Cramton, Y. Shoham, and R. Steinberg (Editors), Combinatorial Auctions, MIT Press, Cambridge, MA, 2005, pp. 507–538.
[4] M.O. Ball, R. Hoffman, D. Knorr, J. Wetherly, and M. Wambsganss, "Assessing the benefits of collaborative decision making in air traffic management," in G. Donohue and A. Zellweger (Editors), Air Transportation Systems Engineering, American Institute of Aeronautics and Astronautics, Inc., Reston, VA., 2001, pp. 239–250 (a preprint appeared in Proceedings of the 3rd USA/Europe Air Traffic Management R&D Seminar, 2000).
[5] M.O. Ball, R. Hoffman, A. Odoni, and R. Rifkin, Efficient solution of a stochastic ground holding problem, Oper Res 51 (2003), 167–171.
[6] M.O. Ball, R. Hoffman, D. Lovell, and A. Mukherjee, "Response mechanisms for dynamic air traffic flow management," in Proceedings of the 6th USA/Europe Air Traffic Management R&D Seminar, Baltimore, MD.
[7] M.O. Ball and G. Lulli, Ground delay programs: Optimizing over the included flight set based on distance, Air Traffic Contr Q 12 (2004), 1–25.
[8] D. Bertsimas and S. Stock, The air traffic flow management problem with enroute capacities, Oper Res 46 (1998), 406–422.
[9] Federal Aviation Administration, FAA Order 7210.3T, available at http://www.faa.gov/atpubs, 2004.
[10] R. Hoffman and M.O. Ball, A comparison of formulations for the single airport ground holding problem with banking constraints, Oper Res 48 (2000), 578–590.
[11] R. Hoffman, Integer Programming Models for Ground-Holding in Air Traffic Flow Management, Ph.D. Dissertation, University of Maryland, 1997.
[12] K. Howard, Meeting minutes of the CDM Collaborative Routing Subgroup, available at http://cdm.metronaviation.com/cr/crmins.html.
[13] K. Howard, ETMS/ATMS system requirements: Slot credit substitutions (SCS) version 1.7, The John A. Volpe Res. Ctr., Cambridge, MA, available at http://cdm.metronaviation.com/ad/CDM-GDP_specs.htm, 2003.
[14] J.P. Ignizio, Linear Programming in Single- and Multiple-Objective Systems, Prentice-Hall, 1982.
[15] H. Luss, On equitable resource allocation problems: A lexicographic minimax approach, Oper Res 47 (1999), 461–378.
[16] A.R. Odoni, "The flow management problem in air traffic control," in Flow Control of Congested Networks, A.R. Odoni, L. Bianco, and G. Szego (Editors), Springer-Verlag, Berlin, pp. 269–288.

[17] O. Richetta and A.R. Odoni, Solving optimally the static ground holding policy problem in air traffic control, Transport Sci 24 (1993), 228–238.
[18] H.D. Sherali and A.L. Soyster, Preemptive and nonpreemptive multiobjective programming: Relationships and counterexamples, J Optim Theor Appl 39 (1983), 173–186.
[19] T. Vossen, Fair Allocation Methods in Air Traffic Management, Ph.D. Dissertation, University of Maryland, Baltimore, MD, 2002.
[20] T. Vossen and M.O. Ball, Slot trading opportunities in collaborative ground delay programs, Transport Sci, in press (2005).
[21] T. Vossen, M.O. Ball, R. Hoffman, and M. Wambsganns, A general approach to equity in traffic flow management and its application to mitigating exemption bias in ground delay programs, Air Traffic Contr Q, 11 (2003), 277–292 (a preprint was published in Proceedings of the 5th EUROCONTROL/FAA ATM R&D Seminar, Budapest, Hungary).
[22] H.P. Young, Equity in Theory and Practice, Princeton University Press, 1994.
[23] M. Wambsganns, Collaborative decision making through dynamic information transfer, Air Traffic Contr Q 4 (1996), 107–123.