

Equitable Allocation of Enroute Airspace Resources

Nasim Vakili Pourtaklo

School of Electrical and Computer Engineering
and Institute for Systems Research
University of Maryland
College Park, Maryland 20742
Email: nasimv@umd.edu

Michael Ball

Robert H Smith School of Business
and Institute for Systems Research
University of Maryland
College Park, Maryland 20742
Email: mball@umd.edu

Abstract—Ration-by-schedule (RBS) has provided a widely accepted resource-rationing principle for ground delay program (GDP) planning in the U.S. Rationing of airspace resources poses significant new challenges not well-addressed by RBS. In this paper, we describe new resource rationing principles and a new methodology for use in rationing access to constrained enroute airspace. While RBS implicitly assumes that all flights requesting a slot must receive one, our new methods explicitly allow some flights to be refused access, since flight operators have the option of rerouting around the constrained airspace. Unlike RBS our methodology requires and makes use of flight operator preference information and it employs randomization. Our methods have potential usefulness both in airspace flow program (AFP) planning and in the emerging System Enhancements for Versatile Electronic Negotiation (SEVEN).

Keywords—resource rationing; flow management; fairness; equitable allocation; AFP

I. INTRODUCTION

The Federal Aviation Administration (FAA) traffic flow management systems have responsibility to respond to predicted capacity-demand imbalances within the U.S. National Airspace System (NAS). Using concepts of Collaborative Decision Making (CDM) (see [23] and [24] for background), a highly successful set of tools and procedures have been deployed and refined over the past 10 years to plan and manage ground delay programs (GDP's). GDP's were originally designed to respond to reductions in the arrival capacity at an airport and this certainly has been their principal application over the years. However, at times, they have also been used to reduce demand within portions of the enroute airspace that are in danger of becoming congested. Most agreed that this was a poor use of GDP tools so a new traffic management initiative, the airspace flow program (AFP)[16], was developed along with an associated set of modeling tools and operational procedures.

The first step in initiating an AFP is the definition of a Flow Constrained Area (FCA), which has both geographic boundaries and temporal boundaries. In general, the geographic domain of an FCA is defined as a volume of enroute airspace, but, it is also quite typical for an FCA to be defined relative to a boundary curve. AFP tools can then identify a list of flights expected to pass through into the FCA. This list, updated with fresh information every five minutes, is sent to the

Flight Schedule Monitor (FSM), which displays the projected demand in a number of formats designed to support effective planning. The traffic flow management (TFM) specialists at the air traffic control systems command center (ATCSCC) enter an FCA capacity, expressed as the number of flights that can be managed per hour, and FSM will then assign each flight a controlled departure time so that the flow into FCA does not exceed the declared capacity. These departure times are sent to the customers for their planning and to the towers at the departure airports for enforcement.

The principal difference between the GDP tools and the AFP tools is that GDP tools can only apply delays to a subset of flights destined for a single airport while the AFP tools can apply delays to a subset of flights predicted to fly through a designated FCA. Otherwise, to a very large extent, GDP concepts have been carried over to AFP's. In this paper, we propose a new method for assigning AFP slots to flights and flight operators, which is fundamentally different from the method currently used for GDP's and AFP's, ration-by-schedule (RBS).

The method we propose applies to a general class of airspace resource allocation problems and, in fact, we have designed it to also be applicable to the emerging System Enhancements for Versatile Electronic Negotiation (SEVEN) [14]. While SEVEN should potentially have a broad range of application contexts, the key feature that it brings to bear that is not present in AFP's, is the ability for flight operators to express preferences among various options for the disposition of an individual flight. The ability for flight operators to express preferences is also a key feature of our proposed resource allocation method.

There are also other important differences between resource allocation for GDP's and enroute resource allocation. In the GDP setting, demand is established based on the set of flights scheduled to arrive at the GDP airport. Since the authority to cancel a flight rests with the flight operator and not the FAA, GDP planning procedures must allocate a slot to all flights within the GDP demand list. Of course, in severe situations, the FAA will be forced to assign extreme flight delays, which may de facto necessitate the cancelation of certain flights. However, GDP procedures implicitly assume all flights must be assigned an arrival slot. On the other hand,

in the AFP or SEVEN setting, all flights on the demand list need not be granted access to the enroute resource. The flight operator has the prerogative to cancel flights not given access *or reroute such flights around the restricted airspace*. Thus, enroute resource allocation decision models must both determine which flights gain access and assign an access time (slot) for those flights that do gain access.

Another related difference is the existence of a fixed flight schedule on which to base resource allocation for GDP's. In fact, as implied by its name, *ration-by-schedule* uses, in a very fundamental way, the flight schedule as the basis for resource allocation. In concept this can be done for enroute problems by simply taking the schedule associated with the list of flights whose flight plans have been filed through the impacted enroute resource. A key difference, however, is that the filing a flight plan is a short-term action and, as a result, the possibility of flight operators trying to "game the system", e.g. by filing unnatural flight plans, is a very real possibility.

Our work uses as a starting point research on GDP's [24] and the investigation of RBS as a basis for fair resource allocation [23]. However, there is an old and extensive literature on fair resource allocation. The problem of sharing "fairly" a given amount of resources is perhaps the oldest one faced by the economists. Brams' books ([6], [7]) are full of examples about divisions of goods. The fair division problem is simply stated in general terms: given a set Ω , given n individuals and given some fairness requirements, find an opportune division. The challenge of dividing indivisible goods has been studied in the literature [5], [8], [2]. The various models of the rationing problem have been addressed in [13], [10] for discrete items. Allocating resources in proportion to individual claims is the oldest formal rule of distributive justice. In case of indivisibility, the probabilistic rationing method gives an expected share to an agent proportional to his claim [17], [21]. In the proportional random allocation method, the assumption is all items are homogeneous. Two simple scheduling methods are discussed in the queuing literature. The *proportional* method amounts to treat equally each unit of claim [19]. In other words, the t -th units goes to an agent with a probability proportional to unsatisfied demand. The *fair queuing* method solves this problem by allocating one unit per agent, irrespective of the size of individual demand, in a successive round-robin fashion. In each round the active agents (whose demands is not yet fully met) are randomly ordered (with uniform probability) and served one job in that order [18]. The problem of fair division when agents have heterogenous preferences over the objects is studied in [20] [1], [4], [11], [12],[15], [3]. In those studies a probabilistic to problem of assigning objects to the agents is suggested. In our problem, we face two difficulties. First, when the FAA as a coordinator must decide about each carrier's share, the underlying good, a time-slot, is not homogeneous in nature. The second difficulty is how to include carriers' preference in the assignment problem. In the following section we address these two difficulties while we try to meet fairness principles.

II. PROBLEM DESCRIPTION AND OVERVIEW OF PROCEDURE

As discussed above the starting point for AFP planning is the definition of an FCA and associated flight list. The simple FCA capacity model employed allows the FCA to be characterized by a set of entry slots, denoted by $\mathcal{S} = \{s_1, s_2, \dots, s_m\}$. We also denote the set of involved flights by $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$. We assume each slot can be used by a single flight and that the flight schedule and status is such that all slots may be assigned to some flight. However, in general $n > m$, i.e. the number of flights is greater than the number of slots. The problem we address is to choose m out of n flights and to assign each of these flights to a unique slot. We also define a set of flight operators, $\mathcal{A} = \{A_1, A_2, \dots, A_K\}$, where each flight operator A_i "owns" a subset of flights, \mathcal{F}_i . As with GDP planning, although flights are assigned to slots, we view the flight-to-slot assignment as a slot-to-flight operator assignment. Moreover, the assignment process takes into account flight operator properties.

We break the process down into two steps:

Step 1: Determine a fair share, FS_i for each flight operator, A_i .

Step 2: Allocate flights to slots in a manner consistent with the fair share determined in Step 1.

Our process employs randomization so that its outcome depends on the draw of random numbers and the solution produced given identical inputs will vary depending on the draw of the random numbers. Ideally FS_i should be the *expected* number of slots that flight operator i receives and, since all slots are always used, we have that the total over all fair shares adds up to the total number of slots, i.e. $\sum_i^m FS_i = m$.

It is probably important at this point to comment on the use of randomization. Randomization is commonly used in the fair allocation literature, but has never been employed formally in practice within ATFM. At CDM meetings airlines have generally expressed distaste for procedures with uncertain outcomes (probably since ATFM is already fraught with uncertainty due to weather and other causes). In this context we feel the use of randomization is justified since it is quite likely that many flight operators, most notably business jet and GA operators, will own a very small number of flights involved in the allocation process. In fact, it is likely multiple flight operators will have only one flight. In this case, FS_i will be less than one. Since in general there could be many more slots than flights, by necessity such flight operators much be denied a slot in some executions of the procedure. The only way such flight operators can receive their fair share is over multiple executions of the procedure, where on some executions they receive no slots, i.e. on some days that flight operator would not receive a slot but on others it would. Randomization with an appropriate expected value criterion is one way to achieve this. Another way is via a "credit system" where credits are

carried over from one day to the next. Such systems have been proposed and certainly have merit; however, they also have not been accepted in practice and have their own set of challenges.

The flight-to-slot assignment carried out in Step 2 is a type of randomized round-robin that employs flight-operator preferences. Each flight operator specifies an ordered list of flight-to-slot assignments. At each iteration, when a flight operator has its “turn”, the the highest available assignment on that flight operator’s preference list is chosen. Here, by available, we mean the the associated flight has not yet been assigned a slot and the associated slot has not been assigned to a flight.

In Section III, we describe the procedure for determining FS_i , i.e. Step1 and Section IV covers Step 2. Section V provides our experimental results.

III. DETERMINING TOTAL SHARE OF AVAILABLE SLOTS FOR EACH CARRIER

As discussed above the goal of this section is to determine a fair share of available slots “owed” to each operator in expectation. Our procedure for determining this fair share requires as input a flight schedule. This schedule can be interpreted as the times at which flights are “scheduled” to arrive at the boundary of the FCA under consideration. Such a flight schedule can be derived based on each flight’s scheduled departure times and filed flight plan. As discussed earlier, employing such a schedule can be problematic as it is not immune to “gaming” or “strategic behavior” on the part of flight operators in order to impact the calculation of their fair share. We recognize this potential flaw and leave as a future research topic the development of strategy-proof schedules or more direct strategy proof fair share calculators. At the same time we believe the analysis we now describe provides a research contribution in the form of method for converting a schedule into flight operator fair shares. Given that current AFP practice bases its allocation on a schedule like the one we employ, our methods can be seen to provide an enhancement to currently used methods, albeit with room still left for improvement.

The availability of a schedule is characterized by knowing for each flight f , a scheduled arrival time a_f , which is interpreted as the time f is “scheduled” to arrive at the FCA boundary. Each slot s_j has an associated time t_j so that a flight f can be assigned to slot s_j if $a_f \leq t_j$.

The flight-to-slot allocation procedure presented in the next section employs flight operator preferences. In this section, to determine each flight operator’s fair share we consider a randomized fair allocation procedure that treats all flights and slots equally in allocating flights to slots. This procedure is based on the following two principles:

- Each slot must be used.
- All flights have equal share of each slot that they can use.

We use the allocation procedure *proportional random allocation*, PRA, which is based on these principles (we note that variants of PRA have been defined and used previously e.g. in [22]).

PRA:

- Step 1** : Set $F_1 = \{f \in \mathbf{F} : a_f \leq t_1\}$ and $i = 1$
 - Step 2** : Choose an $f \in F_i$ with probability $\frac{1}{|F_i|}$ and assign f to s_i
 - Step 3** : Set $i = i + 1$
 - Step 4** : Set $F_i = \{f \in \mathbf{F} : a_f \leq t_i\} - \{f\}$
 - Step 5** : If $i \leq m$ Then go to Step 2.
- End.

Note that PRA uses the two principles given above to randomly assign each flight to a slot.

We define for each flight f and slot s , P_{fs} to be the probability that PRA assigns f to s . Also, define:

$$P_f = \sum_s P_{fs} = \text{PRA share for flight } f \quad (1)$$

$$FS_i = \sum_{f \in \mathcal{F}_i} P_f = \text{PRA share for flight operator } A_i \quad (2)$$

Thus, we see PRA viewed as a random process with associated probability gives a basis for computing FS_i , which was the goal of this section. Of course, it is important to justify why the PRA share for operator, A_i , is an appropriate value for each flight operator’s fair share. We will do this later in this in subsection III-B, but first, we show that the FS_i ’s can be easily computed.

A. Computing FS_i

To derive a formula, we first define:

$$n_i = \text{number of flights that can be assigned to } s_i \\ \text{for } 0 \leq i \leq n - 1.$$

Note that $n_{i+1} \geq n_i$ for $0 \leq i \leq n - 1$, since if a flight can be assigned to s_i , it can be assigned to s_{i+1} . Now, we note that the number of choices that PRA has for assigning a flight to slot s_i is $n_i - (i - 1)$ since there are n_i flights that could be assigned but $(i - 1)$ have already been assigned in the $(i - 1)$ earlier iterations. Thus, the total number of possible solutions that could be outcome by PRA is: $\prod_{i=1}^n (n_i - (i - 1))$.

Now let us consider the number of possible solutions that contain a given flight f , where we define $i(f)$ to be the i such that s_i is the earliest slot to which f can be assigned, i.e. the earliest s_i such that $a_f \leq t_i$. We start by computing the number of solutions that do not contain f and then, knowing that all others do contain f will be able to compute P_f . Using the same logic as above, if f is deleted, then the number of choices for PRA for slots s_i with $i < i(f)$ is $n - (i - 1)$; for $i \geq i(f)$, the deletion of f reduces the number of choices by 1 so the number of possible choices for PRA is $n_i - 1 - (i - 1) = n_i - i$. While our assumption that all slots can be used implies that $n_i - (i - 1) \geq 1$ for all $i \geq i(f)$, it is possible that $n_i - i = 0$ for some $i \geq i(f)$. If this occurs, then the deletion of f implies not all slots can be used and so f must be in all solutions that use all slots. Thus, $P_f = 1$. If $n_i - i \geq 1$ for all

$i \geq i(f)$, then we have that

$$\begin{aligned} P_f &= 1 - \frac{\prod_{i=1}^{i(f)-1} (n_i - (i-1)) \prod_{i=i(f)}^n (n_i - i)}{\prod_{i=1}^n (n_i - (i-1))} \\ &= 1 - \frac{\prod_{i=i(f)}^n (n_i - i)}{\prod_{i=i(f)}^n (n_i - (i-1))} \end{aligned}$$

Given that P_f can be readily computed, we now have a direct efficient way of computing FS_i based on (2).

B. Properties of FS_i

We start with an example. Consider the following set of slots and flights.

Slot:	s_1	s_2	s_3	s_4		
Time:	8:00	8:04	8:08	8:12		
Flight:	A101	B201	A102	A103	B202	C301
Airline	A	B	A	A	B	C
a_f	7:55	8:02	8:03	8:05	8:07	8:10
$i(f)$	1	2	2	3	3	4

Using the definitions above we can calculate:

Result of Application of PRA:

$$\begin{aligned} n_1 &= 1, n_2 = 3, n_3 = 5, n_4 = 6 \\ \text{numb of solutions} &= (1-0)(3-1)(5-2)(6-3) = 18 \\ P_{A101} &= 1 \text{ since } n_1 - 1 = 0 \\ P_{B201} &= P_{A102} = 1 - \frac{(3-2)(5-3)(6-4)}{(3-1)(5-2)(6-3)} = \frac{7}{9} \\ P_{A103} &= P_{B202} = 1 - \frac{(5-3)(6-4)}{(5-2)(6-3)} = \frac{5}{9} \\ P_{C301} &= 1 - \frac{6-4}{6-3} = \frac{1}{3} \\ FS_A &= P_{A101} + P_{A102} + P_{A103} = 1 + \frac{7}{9} + \frac{5}{9} = 2\frac{1}{3} \\ FS_B &= P_{B201} + P_{B202} = \frac{7}{9} + \frac{5}{9} = 1\frac{1}{3} \\ FS_C &= P_{C301} = \frac{1}{3} \end{aligned}$$

It is instructive to compare the PRA-based shares with those produced by RBS. RBS would order all flight based on a_f and allocate slots in order until all slots were exhausted. Thus, we would have:

Result of Application of RBS:

Flight-to-slot assignment:

$$\begin{aligned} A101 &\rightarrow s_1; B201 \rightarrow s_2; \\ A102 &\rightarrow s_3; A103 \rightarrow s_4; \\ B202, C301 &: \text{no slots assigned} \end{aligned}$$

The corresponding flight operator shares:

$$FS_A = 3, FS_B = 1, FS_C = 0$$

We see that the PRA-based approach explicitly takes into account that there are a limited number of slots, i.e. $n < m$. Thus, it gives a share to all involved flights. At the same time it give a larger share to earlier flights, recognizing that earlier flights should receive some benefit relative to later

flights. RBS, on the other hand, implicitly assumes there will be resources to give to all flights, and, when it runs out is not able to give the last two flights anything. Thus, flight operator C receives zero share. While RBS is based on the principle of first-come-first-served, and PRA is related to the classic principle of proportional allocation, which is used when allocated a limited resource among multiple claimants. For example, if there were a single slot, which all flight could access, where airline A had 3 flights, airline B 2 and airline C 1, then the PRA-based allocation would give A 1/2, B 1/3 and C 1/6, which is exactly the proportional allocation. Of course, since we are dealing with discrete resources, slots, having an allocation that results in fractional values is only worthwhile if there is a method for doing a discrete allocation in a manner consistent with the fractional shares. This is the topic of the next section.

We can show that the PRA meets three fundamental principles of the equity which are *impartiality*, *consistency* and *Equal Treatment of Equals*. *Impartiality* states that allocation rule should not discriminate among the flights *except* insofar as they differ in type. In other words, if two flights are indifferent in type and in the feasible set, they will receive the same slot shares. The consistency property states that the expected slot shares should be independent of the order in which flights are assigned to the slots. The last equity principle, *Equal Treatment of Equals* states that if two flight operators have the same schedule, they will receive the same share from available slots.

IV. ASSIGNING FLIGHTS TO SLOTS

As outlined earlier our flight assignment algorithm is a type of round robin that is based on flight operator preferences. We first describe the format of the preferences. Recall that flight operator A_i ‘owns’ a set of flight \mathcal{F}_i . Also recall that the set of feasible assignments for flight f are $\{(f, s_{i(f)}), (f, s_{i(f)+1}), \dots, (f, s_n)\}$. Now a flight operator expresses its preferences by specifying an ordered list of all its feasible flight-to-slot assignments. At each iteration when it is flight operator A_i 's turn, the algorithm makes the highest flight-to-slot assignment *available* on operator A_i 's list. Here we say that a flight-to-slot assignment, (f, s) is available if f has not been assigned to a slot and if no flight has been assigned to s . The following example illustrates a preference list for operator A :

Slot:	s_1	s_2	s_3	s_4	s_5	s_6
Flights:	A01	A02	A03			
$i(f)$	s_1	s_4	s_6			

Preference List for A

1	(A01, s_1)
2	(A01, s_2)
3	(A02, s_4)
4	(A01, s_3)
5	(A01, s_4)
6	(A01, s_5)
7	(A02, s_5)
8	(A01, s_6)
9	(A02, s_6)
10	(A03, s_6)

Note that flight operator A 's highest preference is the earliest slot for its earliest flight, A01. However, if A does not get s_1 or s_2 , then it would prefer to get s_4 so that it could first accommodate A02 as early as possible. Afterward, its preference are consistent with having its earliest flight depart as early as possible.

It might appear that such a preference list is long and would be rather burdensome for a flight operator to produce. However, it would be fairly easy to devise shorthand notations, standard rules and defaults that could greatly reduce the information requirements. For example a fairly standard preference would be for a carrier to specify that its earliest unassigned flight should receive the earliest available slot. There could be a shorthand for this. A slightly more general rule would be to specify a prioritized list of flights only, with the added stipulation that the highest unassigned flight should receive the earliest available flight. Again, there could be a shorthand or standard rule for this.

There are some key properties we seek for an allocation algorithm. Achieving them all can be a challenge so choosing the most appropriate algorithm can involve compromises. We first list the properties and present a recommended algorithm.

Desirable properties of allocation algorithms:

- 1) **The number of slots allocated to a carrier A_i should be between $\lfloor P_i \rfloor$ and $\lceil P_i \rceil$.** This property, which is a version of "equal treatment of equals" is probably the most fundamental to consider. It says that each flight operator should get its fare share (within a tolerance). A related property that is neither stronger nor weaker states that the *expected* number of slots allocated to carrier A_i should equal P_i . It turns out that enforcing one or the other of these properties is fairly easy but enforcing both in general can be challenging.
- 2) **Each flight operator should be motivated to submit a "truthful" preference list.** This is considered a fundamental property of allocation methods, more formally known as *strategy proofness*. If the "dominant" strategy for each flight operator is to submit a its true priority list, then flight operators need not seek to "game the system" and so the problem they face is relatively straight forward. Further, the overall system will be more stable in the sense that there should not be claim that certain operators gained an unfair advantage.
- 3) **The allocation process should be consistent with the flight operator priorities.** Here, consistent is intention-

ally left vague. The process must work within several constraints – most notable the facts the each flight can only use certain slots and that all slots should be used.

Our allocation algorithm, *Preference-Based Proportional Random Allocation* (PBPR), has two phases. It starts by considering the fractional part of each P_i . In Phase 1, carriers are chosen randomly in proportion to these fractional parts. When a carrier is chosen, it is assigned the highest flight-to-slot assignment on its priority list. Each flight operator is assigned at most one slot during this phase. Small flight operators with $P_i < 1$ are only considered in this phase. The second phase also uses a randomized procedure where the remaining slots are considered from earliest to latest. Flight operators, who can use the slot in question, are chosen randomly in proportion to the integer part of the P_i 's.

We now define our allocation algorithm:

BPBPA:

Inputs: Carriers: A_1, A_2, \dots, A_K ,

Carrier Shares: P_1, P_2, \dots, P_K ,

Carrier Preference Lists: $PList_1, PList_2, \dots, PList_K$

Step 0: Calculate $P_1^F, P_2^F, \dots, P_K^F$ and $P_1^I, P_2^I, \dots, P_K^I$, the fractional parts and integer parts of P_1, P_2, \dots, P_K ; set $N_{fract} = \sum i P_i^F$; if $N_{fract} > 0$ proceed to Step 1; otherwise go to Step 2.

Step 1: PHASE 1

Step 1a: from among all carriers A_i with $P_i^F > 0$ choose A_{i^*} randomly in proportion to the value of P_i^F .

Step 1b: Let (f', s') be the highest priority assignment on $PList_{i^*}$. Assign f' to s' and $P_{i^*}^F = 0$

Step 1c: Delete all assignments of the form $(f', *)$ from $PList_i$; delete all assignments of the form $(*, s')$ from all lists $PList_k$ for $k \neq i$,

Step 1d: Set $N_{fract} = N_{fract} - 1$; If $N_{fract} = 0$ then proceed to step 2; otherwise go to Repeat Step 1.

Step 2: PHASE 2

Step 2a: Let s' be the earliest unassigned slot. If no flights can be assigned to s' , then delete s' and skip to Step 2d. Otherwise, from among all carriers A_i with $P_i > 0$ that can use s' , choose A_{i^*} randomly in proportion to the value of P_i .

Step 2b: Let (f', s') be the highest priority assignment on $PList_{i^*}$. Assign f' to s' .

Step 2c: set $P_i = P_i - 1$; delete all assignments of the form $(f', *)$ from $PList_i$; delete all assignments of the form $(*, s')$ from all lists $PList_k$ for $k \neq i$,

Step 2d: If all slots have been assigned then stop; otherwise repeat Step 2.

Let us now consider how BPBPA performs relative to the desirable properties we laid out earlier. First we note that if no slots go unused, i.e. are deleted in Step 2a, then it can be seen that in Step 2 each flight operator A_i with $P_i \geq 1$ (and $P_i^I \geq 1$) receives exactly P_i^I slots. It then immediately follows that BPBPA achieves property 1. This conclusion does depend on no slot being deleted in Step 2a. In fact, one reason we chose to allocate flights-to-slots in order of increasing slot time was

to make it extremely unlikely that any slots would go unused. While this did happen in other allocation methods we tested, it never happened using PBPR. We note that PBPR does not in general achieve the expected value criterion mentioned in Step 1. However, by choosing flight operators in proportion to P_i^F or P_i^I , we get very close to achieving this property as our experiments demonstrate.

We have not formally proved that PBPR achieves strategy proofness (Property 2). However, it can easily be seen that in Phase 2, no advantage can be gained by reordering elements in the preference list since this will not impact whether a flight operator receives a slot – only which flight is allocated to the slot obtained. There is some potential for gaming when one considers the interaction between Phases 1 and 2, but we feel such possibilities are very minor since most slots (in problems of realistic size) will be allocated in Phase 2.

Finally, we consider Property 3, which simply says that the allocation should be consistent with the preferences. As stated, this is a very weak requirement. Certainly the preferences are taken into account and impact the allocation. At the same time, we should note that preferences would play a greater role if Phase 2 were executed as Phase 1 is, i.e. so that whenever a flight operator were chosen they could choose any available slot. We experimented with such an approach but ran into problems with slots going unused at the end, i.e. slots would have to be deleted in Step 2a.

V. EXPERIMENTAL RESULTS

For our experiment, we used a test data set that had been employed by the CDM Future Concepts Team to perform human-in-the-loop experiments related to SEVEN. It contained 386 flights with 38 flight operators. The data included scheduled arrival times at an FCA boundary. The FCA duration was from 18:00 pm to 21:00 pm. We generated a flight cost function that is described below. Given the cost function, we generated the priority list for each flight operator based on the following principle:

Of all available flights that could use a slot, the flight operator preferred allocating the slot to the flight with the highest marginal cost of delay.

We consider the cost of each flight as a function of number of passengers (actually number of aircraft seats) and the flight delay. The cost function (based on generic advice from airline dispatchers) was constructed based on the following general principles. First, the initial 15 minutes of delay is considered free. The Air Transportation Association (ATA) cites that direct operating cost during block time is \$64 per minute. Ground cost to be about 1/2 as expensive as air cost (and this is accepted practice in the literature), so $\$64/2 = \32 . We can assume that there is also the possibility of rerouting the flight. This effectively caps the delay cost (once the delay cost curve exceeds the rerouting cost, the airline is better off rerouting the flight). As with the airline cost, we have assumed that passengers are willing to ignore the first 15 minutes of delay, that their time is worth \$0.6 for each minute thereafter, and that this linear function is capped after 15 hours[9]. The 0.6

figure again comes from the ATA web site, which cites \$34.88 per hour as an average cost of passenger time. This translates to $\$34.88/60 = \0.5813 per minute, per passenger, which we rounded to \$0.6 per minute. Our cost function should represent internal airline costs. Airlines are interested in provided good customer service but don't suffer the full brunt of passenger costs. We approximated this customer service perspective by multiplying the passenger cost function by 1/6 and adding the resultant cost to the flight delay costs as described above. Thus, we can write the flight delay cost function as:

$$C(x, P) = \begin{cases} 0 & x \leq 15 \\ (32 + 0.1P)(x - 15) & 15 < x \leq M_p \\ (32 + 0.1P)(M_p - 15) & x > M_p \end{cases}$$

Where M_p is flight specific max delay. That is, it is assumed that after M_p minutes of delay, the flight operator would prefer to reroute the flight. Since the cost effectiveness of rerouting will vary with flight characteristics we chose M_p randomly with uniform likelihood between 30 to 90 minutes.

We compared the results of PBPR against ration-by-schedule (RBS), which is currently used to allocate FCA access during airspace flow programs. Our version of RBS proceeded from the earliest to latest slot. At each step, it assigned the available flight with the earliest scheduled arrival time (ties were broken randomly with equal likelihood). Once we determined a flight-to-airline assignment, if multiple flights from the chosen airline could be assigned to the same slot, then we assigned the flight with the highest marginal cost of delay. In this way, at the end of the procedure, the airlines could not improve their cost function by doing an "internal" flight-to-slot reassignment.

In our experiment, we considered 40%, 50%, 60%, 70% and 80% en-route capacity reduction for the FCA. We performed 2000 repetitions of the procedure (note that since PBPR uses randomization its "expected" impacted can only be calculated by doing multiple repetitions). The following table shows the average number of slots assigned to each airline for a specific capacity reduction.

A noteworthy point to be made is that many airlines received 0 slots under RBS (even with 2000 repetitions). This, of course, is to be expected since RBS is a deterministic procedure. Such airlines had only a single flight demanding access to the FCA and that flight had a relatively late scheduled arrival time. On the other hand, the fractional values achieved by PBPR indicate that on some repetitions PBPR allocated such an airline a slot and on others it did not. Few would probably dispute that this is a more equitable outcome.

An important related issue is the degree to which PBPR achieves (on the average) the flight operator fair shares (P_i 's)(Labeled Total Share in the Table I). As stated earlier we cannot formally prove that this is the case. As the results in the table indicates experimentally PBPR comes very close achieving P_i values. As would be expected, the RBS can diverge by fairly significant amounts.

Of course, a very fundamental implicit goal of our procedure is that flight operators should be able to improve their internal

Airlines	Number of Slot Allocated to Each Flight Operator, RBS vs. PBPRA														
	80% Capacity Reduction			70% Capacity Reduction			60% Capacity Reduction			50% Capacity Reduction			40% Capacity Reduction		
	PBPRA	RBS	Total Share	PBPRA	RBS	Total Share	PBPRA	RBS	Total Share	PBPRA	RBS	Total Share	PBPRA	RBS	Total Share
1	5.112	6	5.079	7.346	8	7.3	9.36	10	9.32	11	11	11.19	12.8	13	13
2	1.603	1	1.553	2.25	3	2.2	2.709	4	2.82	3.5	4	3.407	3.79	4	4
3	0.597	0	0.58	0.738	1	0.9	1.308	2	1.27	1.6	2	1.653	2.09	2	2.1
4	0.154	0	0.107	0.226	0	0.2	0.305	0	0.26	0.4	0	0.362	0.55	0	0.5
5	1.625	0	1.673	2.581	4	2.6	3.618	5	3.67	4.7	6	4.788	5.79	6	5.9
6	1.548	2	1.475	2.306	3	2.2	3.088	3	3.06	3.7	3	3.864	4.61	5	4.6
7	0.344	0	0.262	0.463	0	0.4	0.581	0	0.62	0.7	1	0.845	1.17	1	1.1
8	1.774	2	1.998	2.698	2	2.8	3.434	4	3.39	3.8	4	3.937	4.55	5	4.4
9	0.141	0	0.094	0.189	0	0.2	0.284	0	0.23	0.4	0	0.327	0.52	0	0.4
10	2.546	3	2.508	3.712	4	3.8	5.234	4	5.2	6.6	6	6.561	7.77	8	7.9
11	0.29	0	0.21	0.393	0	0.3	0.506	1	0.47	0.6	1	0.621	0.74	1	0.8
12	0.236	0	0.178	0.337	0	0.3	0.454	0	0.4	0.6	1	0.544	0.68	1	0.7
13	8.049	13	8.035	11.688	15	12	15.43	16	15.4	19	17	18.68	21.6	20	22
14	0.784	1	0.994	1.567	2	1.6	2.296	2	2.24	2.8	2	2.952	3.71	3	3.7
15	6.451	7	6.405	9.527	7	9.5	12.6	9	12.6	16	12	15.89	19.3	16	19
16	0.441	0	0.358	0.581	0	0.6	0.712	1	0.84	1.2	1	1.149	1.56	1	1.5
17	0.316	0	0.257	0.435	1	0.4	0.538	1	0.56	0.7	1	0.703	0.74	1	0.8
18	0.022	0	0.017	0.04	0	0	0.061	0	0.04	0.1	0	0.062	0.14	0	0.1
19	0.187	0	0.145	0.295	0	0.2	0.394	0	0.34	0.5	0	0.47	0.65	1	0.6
20	0.034	0	0.027	0.074	0	0	0.102	0	0.07	0.2	0	0.11	0.22	0	0.2
21	11.771	15	11.953	17.689	19	18	23.65	22	23.7	30	28	29.49	34.8	34	35
22	1.74	1	1.868	2.568	2	2.6	3.324	4	3.28	4.1	4	4.036	4.76	5	4.8
23	0.293	0	0.23	0.42	0	0.4	0.564	0	0.55	0.7	0	0.766	1.04	1	1
24	1.131	1	1.087	1.63	1	1.7	2.403	2	2.35	3.1	3	3.064	3.73	4	3.8
25	1.662	2	1.697	2.596	4	2.6	3.62	5	3.63	4.6	5	4.655	5.63	6	5.6
26	0.238	0	0.184	0.342	0	0.3	0.442	0	0.41	0.6	1	0.555	0.7	1	0.7
27	0.124	0	0.098	0.202	0	0.2	0.276	0	0.24	0.4	0	0.337	0.53	0	0.5
28	0.764	2	0.924	1.43	2	1.4	1.675	2	1.78	2.1	2	2.091	2.38	2	2.3
29	8.777	10	8.97	13.087	14	13	16.74	18	17	21	23	20.71	24.3	25	24
30	0.509	0	0.48	0.668	1	0.7	1.047	1	1.03	1.4	2	1.328	1.61	2	1.6
31	0.273	0	0.211	0.374	0	0.3	0.486	1	0.46	0.6	1	0.607	0.7	1	0.8
32	8.22	7	8.161	11.719	10	12	15.65	16	15.7	20	18	19.5	23.2	24	23
33	0.073	0	0.049	0.101	0	0.1	0.169	0	0.13	0.3	0	0.183	0.32	0	0.3
34	1.702	1	1.783	2.663	3	2.8	3.682	6	3.81	4.8	6	4.903	5.77	7	5.9
35	0.079	0	0.061	0.138	0	0.1	0.195	0	0.16	0.3	0	0.226	0.4	0	0.3
36	7.563	4	7.512	11.586	10	12	16.15	16	16.1	21	27	20.87	25.7	29	26
37	0.196	0	0.155	0.325	0	0.3	0.419	0	0.36	0.5	1	0.497	0.66	1	0.6
38	0.631	0	0.62	1.017	0	1	1.514	0	1.49	2.1	1	2.077	2.73	2	2.8

TABLE I
COMPARISON OF PBPRA AND RBS ALLOCATIONS

performance based on an allocation process that takes into account their preferences. The total cost saving over all airlines of PBPRA compared to RBS is shown in Fig 1. We note that PBPRA consistently provides a significant savings.

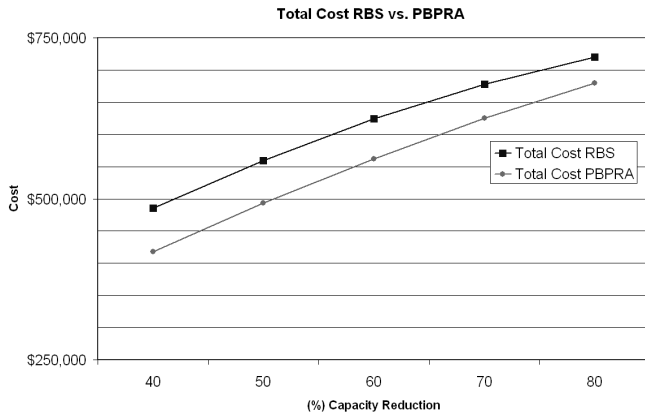


Fig. 1. Total cost of all flight operators, RBS vs. PBPRA.

Table II provides the corresponding percentage savings.

% Capacity reduction	% Total cost saving
40	14.02
50	11.78
60	9.87
70	7.82
80	5.63

TABLE II
TOTAL COST SAVINGS PBPRA COMPARED TO RBS

VI. CONCLUSIONS

In this paper we provided a new approach to allocating limited available airspace resources. Our methods have potential applicability both to the planning of airspace flow programs (AFP's) and to the emerging System Enhancements for Versatile Electronic Negotiation (SEVEN) concepts. We provide two related contributions. The first is a method for computing a fair share for each flight operator and the second is a method for allocating flights to slots in a way that is consistent with the computed fair shares.

We believe that it is necessary to follow this work up with detailed simulations, including human-in-the-loop simulations. These would no doubt lead to refinements to our procedures necessary to implement them in practice. For example, an obvious area to study would be methods for streamlining the preference list. We feel it would also be important to study at a more fundamental level the degree to which PBPRA achieves the list of desirable properties. Also, as stated in the text, the problem of computing a fair share that does not depend on a schedule is clearly an important, challenging research question.

ACKNOWLEDGMENT

The authors would like to thank Dr. Robert Hoffman for providing us with flight data and also his help for computing cost function for the flight operators. This work was supported by the National Center of Excellence for Aviation Operations Research (NEXTOR), under FAA cooperative agreement number 01CUMD1. Opinions expressed herein do not necessarily reflect those of the FAA or the U.S. Department of Transportation.

REFERENCES

- [1] A. Abdulkadiroglu, T. Sonmez. Ordinal efficiency and dominated sets of assignment. *Journal of Economic Theory*, 112:157-172, 2003.
- [2] A. Alkan, G. Demange, and D. Gale. Fair allocation of indivisible goods and criteria of justice. *Econometrica*, 59:1023-1039, 1991.
- [3] A. Bogomolnaia and H. Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100:295-328, 2001.
- [4] A. Bogomolnaia and H. Moulin. Collective choice under dichotomous preferences. *Journal of Economic Theory*, 122:165-184, 2005.
- [5] S. J. Brams. Fair Division. Mimeo, New York University, Department of Political Science, 2005.
- [6] S. J. Brams, and A.D. Taylor. Fair Division: from Cake-Cutting to Dispute Resolution, Cambridge University Press, 1996.
- [7] S. J. Brams, and A.D. Taylor. The Win-Win Solution: Guaranteeing Fair Shares to Everybody. Norton & Co. 1999.
- [8] I. Bezakova and V. Dani. Allocating indivisible goods. *ACM SIGecom Exchanges*, 5:11-18, 2005.

- [9] S. Bratu and C. Barnhart, An analysis of passenger delays using flight operations and passenger booking data, *em Air Traffic Control Quarterly*, 13, 2005.
- [10] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of fair queuing algorithm. *Internetworking: Res Experience*, 1:3-26, 1990.
- [11] L. Ehlers. Probabilistic allocation rules and single-dipped preferences. *Social Choice and Welfare*, 19:325-348, 2002.
- [12] L. Ehlers, H. Peters, and T. Storcken. Strategy-proof probabilistic decision schemes for one dimensional single-peaked preferences. *Journal of economic theory*, 105:408-434.
- [13] J. Elster. *Local Justice*. Russle Sage Foundation, New York, 1992.
- [14] Gaertner, N, M. Klopfenstein and G. Wilmouth, Updated Operational Concept for System Enhancements for Versatile Electronic Negotiation (SEVEN), September 7, 2007 available at <http://cdm.fly.faa.gov/index.html>
- [15] A. Katta and J. Sethurman. A solution to the random assignmnet problem on the full preference domain. *Journal of Economic Theory*, 131:231-250, 2006.
- [16] Metron Aviation. www.metronaviation.com/airspace-programs.php
- [17] H. Moulin. The proportional random allocation of indivisible units. *Social Choice and Welfare*, 19:381-413, 2002.
- [18] H. Moulin and R. Stong. Fair queueing and other probabilistic allocation methods. *Mathematics of Operations Research*, 27:1-30, 2002.
- [19] H. Moulin and R. Stong. Filling a multicolor urn: an axiomatic analysis. *Game and Economic Behavior*, 45:242-269, 2003.
- [20] Y. Sprumont. The division problem with single-peaked preferences: a characterization of uniform allocation rule. *Econometrica*, 59:509-519, 1991.
- [21] A. Tasnadi. On probabilistic rationing methods. *Mathematical Social Science*, 44:211-221, 2002.
- [22] T.W.M. Vossen *Fair Allocation Methods in Air Traffic Management*. PhD Thesis, University of Maryland, R.H. Smith School of Business, 2002.
- [23] Vossen, T., M. Ball. 2006. Optimization and mediated bartering models for ground delay programs. *Naval Research Logistics*, 53:75-90.
- [24] M. Wambsganns. Collaborative decision making through dynamic information transfer. *Air Traffic Control Quarterly*, 4:107-123, 1996.

AUTHOR BIOGRAPHY

Nasim Vakili Pourtaklo is a PhD candidate at university of Maryland where she received her Ms. degree in electrical engineering. She received her Bs. in electrical engineering from Sharif University of Technology. She worked as a Program Manager at Microsoft. She is currently a researcher at the NEXTOR lab where she is working on resource allocation and optimization projects.

Michael Ball is the Orkand Corporation Professor of Management Science in the Robert H. Smith School of Business at the University of Maryland. He also holds a joint appointment within the Institute for Systems Research (ISR) in the Clark School of Engineering. Dr. Ball received his PhD in Operations Research in 1977 from Cornell University. He is co-Director of NEXTOR, the National Center of Excellence for Aviation Operations Research, and he leads the NEXTOR Collaborative Decision Making project.