Stochastic Integer Programming Models for Ground Delay Programs with Weather Uncertainty

Charles N. Glover
Applied Mathematics and Scientific Computation Program & Institute for Systems Research
University of Maryland, College Park, MD 20742

Michael O. Ball
Robert H. Smith School of Business & Institute for Systems Research
University of Maryland, College Park, MD 20742

***Abstract*— Convective weather is a major contributor to air traffic delays. There is much uncertainty associated with weather predictions so stochastic models are necessary to effectively assign ground delay and route adjustments to flights. We describe a two-stage stochastic integer program for this problem. We then compare the results of this formulation to algorithms already in the literature.**

Keywords: air traffic flow management, integer programming. stochastic programming

I. Introduction

A major priority of the Federal Aviation Administration (FAA) is to reduce airport congestion. Air traffic flow management specialists within the FAA seek to address this issue by resolving instances in the National Airspace System (NAS) where the anticipated demand exceeds airport capacity. A limitation on the number of airports that can be built, paired with a significant increase in air traffic leads to airport congestion being a primary concern. Added to the simple increase in air traffic is the more complex situation that convective weather has on causing demand to exceed capacity at airports. When the FAA predicts this to occur at an airport, they are placed into a situation where airport landing slots become a scarce resource, and must be allocated to flights through some traffic flow management initiative. One of the most advanced such procedures is that of a ground delay program (GDP). Rather than assigning delay to flights in the air, a GDP is a preemptive measure that holds aircraft on the ground before they depart their origin airports. The net effect of this is that the more costly and more risky usage of airborne delay is reduced and transferred to the ground where it is more easily managed. This also reduces the stress on air traffic managers, who have limited options once the aircraft are airborne.

GDPs were first implemented after the airline strike of 1981 [6]. Since then, they have become a growing part of our airline industry. In 2006, there were 1305 GDPs implemented in the United States [5]. The cost of these delays to the airlines and passengers is billions of dollars per year. So it is only logical that we would like for this delay to be at a minimum.

There are many roadblocks to efficient minimization of delay in a GDP. One primary such roadblock is that of equity. Before the current standard of Collaborative Decision Making (CDM) was adopted, participants felt that GDPs were implemented in an inequitable manner. Airlines (correctly) felt that, in many cases, the information they provided was used by the FAA to provide a much greater benefit to their competition then to the airline providing the information. Thus, they would provide out of date or inaccurate information. This lack of equity led to inefficient solution procedures and often resulted in more system delay. CDM was initiated to resolve these issues by instituting methods that were based on agreed upon standards and allocation procedures that provided incentives for participation with honest information [1], [9].

One of the major results of this was the ration-by-schedule (RBS) principle, which decoupled the information provided by the airlines on a day of operations and the resources they received. The basics of the RBS principle are first scheduled first served, so in a GDP the flights are kept in the order that they were originally scheduled. Some flights, though, are exempt from RBS. One set, flights that have already taken off, obviously cannot be given ground delay and must be exempt. The other set, though, is not as simple.

Because of the stochastic nature of weather, an air traffic manager is reluctant to delay a flight several hours in advance of a storm that may or may not materialize. An overly pessimistic forecast could result in some longer flights being given what, in hindsight, is unnecessary delay. To offset this, a distance radius is set from the troubled airport, and ground delays are only assigned to flights that originate inside that radius. The remaining flights are exempt from this GDP.

Ball et al. [2] developed a formal stochastic model of GDP’s to gain a fundamental understanding of how giving preferential treatment to long-haul flights improves expected GDP performance. They proposed the ration-by-distance (RBD) algorithm, which allocates flights to arrival time slots using a priority scheme based on flight list ordered by decreasing flight length. This algorithm is structurally the same as RBS, but with an alternative priority scheme. They showed that RBD, under a fairly general model of GDP dynamics, minimizes the expected delay [2]. It is easy to see, however, that RBD can generate an inequitable distribution of flight delays. To address this problem, they proposed a heuristic algorithm, E-RBD, that would seek to balance efficiency and equity. In this paper, we formulate an integer program (IP), which represents the GDP with weather uncertainty. We then show how this IP can use different objective functions to more precisely balance efficiency and equity to a larger scale than either RBS, RBD or E-RBD.

II. Related Work

The GDP is a well-studied problem in aviation research. Odoni first proposed an IP model for the Ground Holding Problem [8]. Bertsimas and Stock Patterson formulated a model to address issues concerned with congestion in the National Airspace System (NAS)[4]. This model minimizes the total ground delay and airborne delay, while ensuring that the departure capacities, arrival capacities, sector capacities and time connectivity constraints are not violated. Although, the model is for the ATFM problem, it can easily be adapted to represent the Single Airport Ground Holding Problem (SAGHP) and Multiple Airport Ground Holding Problem (MAGHP). These models are deterministic and do not account for the ways that the weather uncertainty can play into the planning of a GDP.

Ball et al. studied a stochastic case of GDPs [3]. In this problem, they were concerned with Airport Arrival Rates (AARs), the number of flights the airport can receive in a given time period, in an environment where the weather is uncertain. The model takes into account an AAR distribution, and produces a planned AAR (PAAR) vector, which is the number of flights that the airport should schedule to arrive in each time period, given the stochastic nature of the weather and the probabilities of different AARs. Kotnyek and Richetta showed that a model first proposed by Richetta and Odoni could also be used to produce the PAAR vector [6]. This model is larger in size than the Ball et al model, but because its cost function for ground delay is more general, it allows for more specific adjustments of the relationship between the costs of airborne holding and ground holding.

Both these models operate under the condition of weather uncertainty. Due to the excessive costs of airborne holding when compared to that ground holding, both papers try to avoid the situation where airport has more flights seeking landing than it has landing slots available in a given time period. But it is also possible to have a larger number of available landing slots than flights seeking landing. Such a situation can arise when an airport expects convective weather and flights are given more ground delay than necessary to offset the convective weather. In these situations the airport would like to be able to re-schedule flights to utilize this unexpected capacity. Because the papers by Ball et al. and Kotnyek and Richetta consider only the static case of stochastic ground delay programs, their models do not allow us to adjust the delays once the weather uncertainty becomes certain.

Mukherjee and Hensen presented a dynamic stochastic IP formulation for the SAGHP, which took as part of its input the possible changes the weather can take throughout the duration of the GDP. [7] This formulation presented a scenario tree to capture all the possible changes in weather outcomes. This scenario tree can grow large in size though and can make the IP computationally inefficient.

In [2], Ball et al consider the problem of maximizing the throughput into the airport. Here, the RBD algorithm is first proposed. The authors prove that the RBD algorithm minimizes total expected delay if the GDP cancels earlier than anticipated. In their proof, the authors were able to compare the total expected delay of the RBD allocation with that of other allocations and are able to show optimality.

III. Formulation

The input to the model comes from two sources: flight-based input and airport-based input. The flight-based input includes:

- The length of the flight k, $len(k)$
- The scheduled arrival time of the flight k, $a(k)$
- The arrival slot that the fight k would receive in the RBS allocation, RBS(k)

The airport-based input includes:

- The expected duration of the GDP
- The nominal capacity of the airport, $cap_2(i)$
- The reduced capacity of the airport, $cap_1(j)$
- A list of possible end times for the GDP
- A probability for each possible end time

We also base our model on some important assumptions.

- The weather has only two possible states, clear and not clear. This is done to keep the problem from growing too large and to model how GDPs are handled in practice, where a GDP is not cancelled until the weather is clear. This collapses the scenario tree and allows the problem to be staged as a two stage stochastic IP instead of as a multi-stage stochastic IP.
- There is no layover between weather clearance time and the time the airport goes back to nominal capacity. We assume this happens immediately.
- We do not consider the airborne holding as an alternative option for ground holding. We will thus not allow for solutions that, in expectation of an early weather clearance time, send more flights to the airport at a given arrival slot than that slot will allow.
- We assume no prior knowledge of weather clearance until the weather has actually cleared.

We will formulate the IP as a stochastic IP. This will take place in two stages. In a GDP, every flight must initially be assigned to a slot, and the first stage models these actions. This takes place with no future knowledge of when the weather will clear.

Let $x_{k,i}$ be the binary variable which is non-zero if flight k is initially assigned to the arrival slot i. Then the following two constraint sets model the stage 1 restrictions. These constraints are very similar to the model proposed by Odoni [8].

$$\sum_{i \geq a(k)} x_{k,i} = 1 \text{ for each flight } k, \quad (1)$$

$$\sum_{k|a(k) \leq i} x_{k,i} \leq cap_1(i) \text{ for each arrival slot } i. \quad (2)$$

Each flight has a scheduled arrival time, $a(k)$, and the first constraint set ensures that each flight is assigned to some arrival slot after its scheduled arrival time.

Notice that we place no upper bounds on the latest arrival slot to which a flight can be assigned. We note that deviations from a flights RBS slot, RBS(k) are a measure of inequity. The E-RBD algorithm seeks to find equitable solutions by restricting how long a flight can be delayed after its RBS slot. We can make this adjustment to the model by placing this restriction, RBS(k) + δ, as an upper bound on the summation in (1).

$$\sum_{i \geq a(k)}^{RBS(k)+\delta} x_{k,i} = 1 \text{ for each flight } k, \quad (1a)$$

Each slot has an initial capacity, $cap_1(i)$, the number of flights the airport can handle during the reduced capacity. So the second constraint set ensures that no slot is utilized in excess of its capacity during the GDP.

Because we do not define inequitable solutions as not feasible, this summation is set over all flights whose scheduled arrival time is before the slot time. If we wished to make this disqualify these inequitable solutions we would instead sum over flights whose scheduled arrival time is before the slot time and whose latest possible arrival is after i.

$$\sum_{k|a(k) \leq i \leq latest\_arr(k)} x_{k,i} \leq cap_1(i) \text{ for each arrival slot } i. \quad (2a)$$

This completes stage 1 of the formulation.

In stage 2 we have a scenario, t, for each possible weather clearance time. Upon clearance of the weather, we assume that the number of arrival slots has immediately increased back to full capacity. In such a situation, it is very possible for flights to be assigned to earlier slots than the slot to which they were initially assigned.

The following constraint set sets up a queue in each scenario of stage 2 amongst the slots available in that scenario. The function *earliest(k, i, t)* maps the allocation (k, i) from stage 1 to the earliest arrival slot that it can be reallocated to in scenario t of stage 2. If *earliest(k, i, t) = j*, then the variable $x_{k,i}$ can enter the scenario t queue at slot j, depending on whether its value is 1 or not. The variable $z_{j,t}$ is the amount that is passed from slot j-1 to slot j in scenario t. The following constraint immediately follows:

$$\sum_{(k,i,t)|earliest(k,i,t)=j} x_{k,i} + z_{j-1,t} - z_{j,t} - u_{j,t} = 0 \text{ for each arrival slot } j \text{ and each scenario } t \quad (3)$$

This says that the allocations such that *earliest(k, i, t) = j* will enter the queue at slot j. Allocations that did not leave the queue at slot j-1 are sent to slot j through the variable $z_{j-1,t}$. Those allocations that depart the queue at slot j do so via the variable $u_{j,t}$. There will need to be three different versions of this constraint, depending on whether the slot j is the first slot (in which case, there is not $z_{j-1,t}$ variable), the last slot (in which case, there is no $z_{j,t}$ variable), or an in-between slot.

The flight dependent input to this IP is the arrival time of each flight, $a(k)$, the length of each flight, $len(k)$, and the RBS allocated slot of each flight, RBS(k). Based on this input, we are able to determine *earliest(k, i, t)* for each stage 1 allocation (k, i) as a pre-processing step.

An allocation (k, i) where $x_{k,i}$ = 1 can be in one of three states at the beginning of a stage 2 scenario: It is either in the air, on the ground because its scheduled departure time has not yet passed, or on the ground because it is serving delay. The determination of which of these sets an allocation belongs to consists of checking the delayed departure time of the allocation, $i - len(k)$, and comparing it with both the weather clearance time, t, and the scheduled departure time of the flight, $a(k) - len(k)$. If the delayed departure time is after the weather clearance time, t, then the flight has departed; if it is before $a(k)$, then it is on the ground because its scheduled departure time has not yet passed; and if it is after $a(k)$ and before t, then it is on the ground because it is serving ground delay.

Depending on which state an allocation is in, there are limited recourse actions that can be taken. Flights already in the air cannot depart for an earlier arrival slot than the one which they are initially assigned. The earliest arrival slot these flights can be assigned to is thus the slot to which were initially assigned, so *earliest(k, i, t) = i* for these flights. Flights on the ground because their scheduled departure time has not yet passed still cannot depart. In this scenario though, they will be free to depart for any arrival slot equal to or after their scheduled arrival slot. The earliest arrival slot for these flights is thus their scheduled arrival slot, so *earliest(k, i, t) = a(k)* for these flights. Flights that are grounded because they are serving delay can depart immediately. These flights, though, cannot arrive at an arrival slot earlier than the time it takes to travel from origin to destination, so for these flights *earliest(k, i, t) = t + len(k)*.

We also need to ensure that the nominal capacity is not violated. This can be achieved by the following constraint:

$$u_{j,t} \leq cap_2(j). \quad (4)$$

The variables $x_{k,t}$ are assumed to be binary. All other variables take on nonnegative integer values.

The objective function will consist of two components: one to measure efficiency and one to measure equity. The metric for efficiency is based on minimizing the total expected delay, which for an individual flight can be recorded as the flight's final assigned time minus its scheduled time. We do not keep track of the final assigned times for each flight in this formulation. So, instead, we can measure efficiency by subtracting the sum of the flight scheduled times from the sum of the assigned times. Because we want the expected delay, we also multiply each term by the probability of that scenario occurring. This metric can then be represented by:

$$\sum_{t \in Scenarios} p_t \left(\sum_{j \in Slots} (j \cdot u_{j,t}) - \sum_{k \in Flights} a(k) \right). \tag{5}$$

The metric for equity will be based on the RBS solution being the most equitable solution. Each flight will then be penalized by how much later they are assigned from their RBS allocation. This can be represented as:

$$\sum_{i \in Slots} \sum_{k \in Flights} \mathrm{cost}(k,i) \cdot x_{k,i} \ , \tag{6}$$

where *cost(k, i)* is the deviation of the stage 1 assignment of flight k from its RBS slot, i.e. *cost(k, i)* = 0 if $i \le \mathrm{RBS}(k)$ and cost(k, i) = i – RBS(k) otherwise.

IV. Experimental Results

We tested this formulation using data based on GDPs run on three different dates at San Francisco International Airport (SFO). The tests were run on three different probability distributions: a uniform distribution, a binomial distribution where the probability of clearance increases as the clearance time increases, and a binomial distribution where the probability of clearance decreases as the clearance time increases.

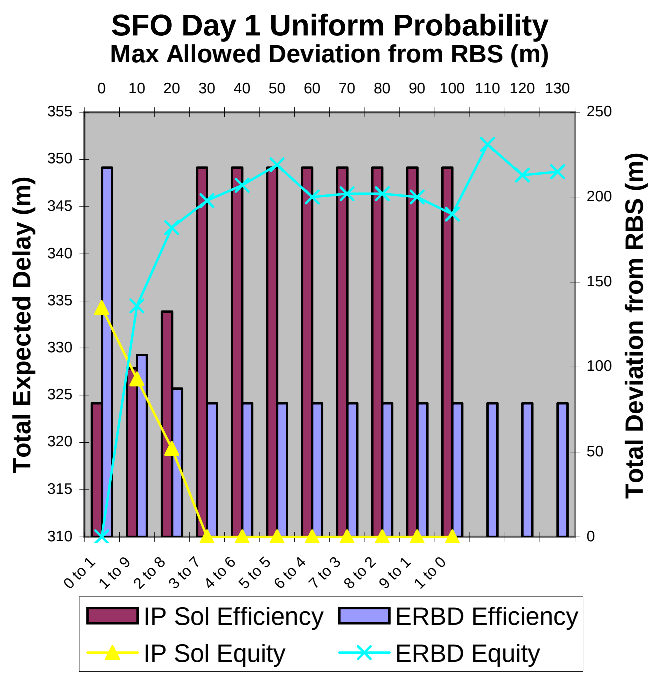
For our nominal capacity we used the Airport Arrival Rate (AAR) of 60 flights per hour, and for our reduced capacity, we used the AAR of 36 flights per hour, or more precisely, we used 6 flights every 10 minutes.

For the IP, the experiment was conducted on each day and distribution with coefficients ranging from 0 to 1, incrementing by 0.1. To obtain the E-RBD results we used an objective function that measured only efficiency and restricted the allowed assignments of a flight to only δ minutes after its RBS allocated slot, for each given δ.

Both the IP and the E-RBD algorithm give the RBS and RBD solutions at their extreme parameter values. For instance, if we set the coefficient for equity to 1 in our IP, our focus is only on equity. We will then receive the most equitable solution, RBS. On the other hand, if we set the coefficient for efficiency to 1 in our IP, our focus is only on efficiency. We will then receive the solution that has the least total expected delay, which is the RBD solution.

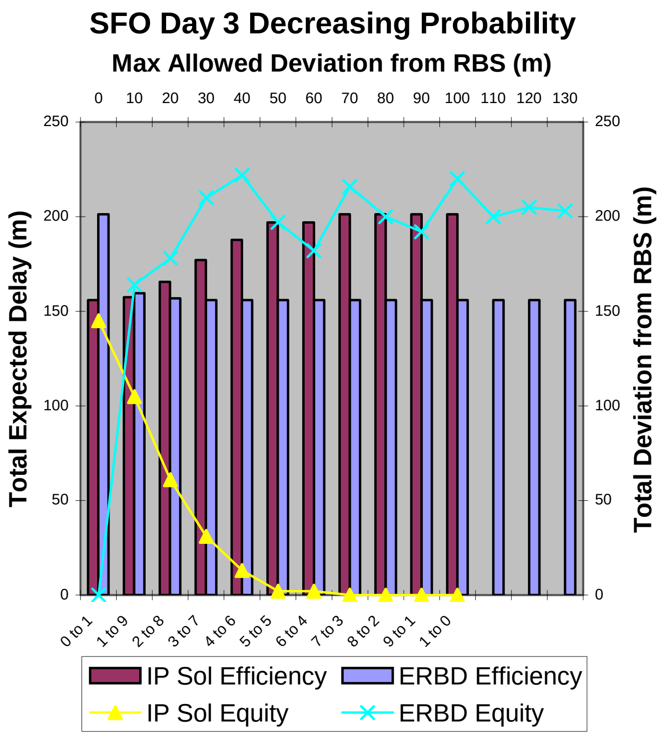
The E-RBD algorithm has similar properties. When we set the maximum deviation, δ, equal to 0, we are not allowed to deviate from the RBS solution, which is the optimal solution in that case. If we set δ to an arbitrarily large constant then all stage 1 assignments are allowed, in which case, the E-RBD algorithm will output the RBD solution.

One key difference between the IP and the E-RBD algorithm, though, is the fact that the IP allows us to choose a coefficient small enough that it keeps us close to either the RBD or RBS solution, while still taking into account both equity and efficiency. For instance, the RBS and RBD solutions obtained from our IP were obtained with equity to efficiency ratios of 1 to 99 and 99 to 1 respectively. This helped find a "more equitable" RBD solution or "more efficient" RBS solution amongst the many available. We did not make such considerations with the E-RBD algorithm because the algorithm, as described in literature, is based on optimizing efficiency by placing a limitation on inequity.



In this first example, we see that with an equity coefficient of 0.1, we are able to obtain an IP solution, which is close in efficiency to the RBD solution, and is more equitable than any of the solutions returned by the E-RBD algorithm, with the exception of the RBS algorithm. The IP solution with an equity coefficient of 0.2 is also more efficient than the RBS solution.

Neither the IP or the E-RBD algorithm give many solutions outside of RBD and RBS. This leads to an immediate question of whether this is true in general, or just a product of this example.

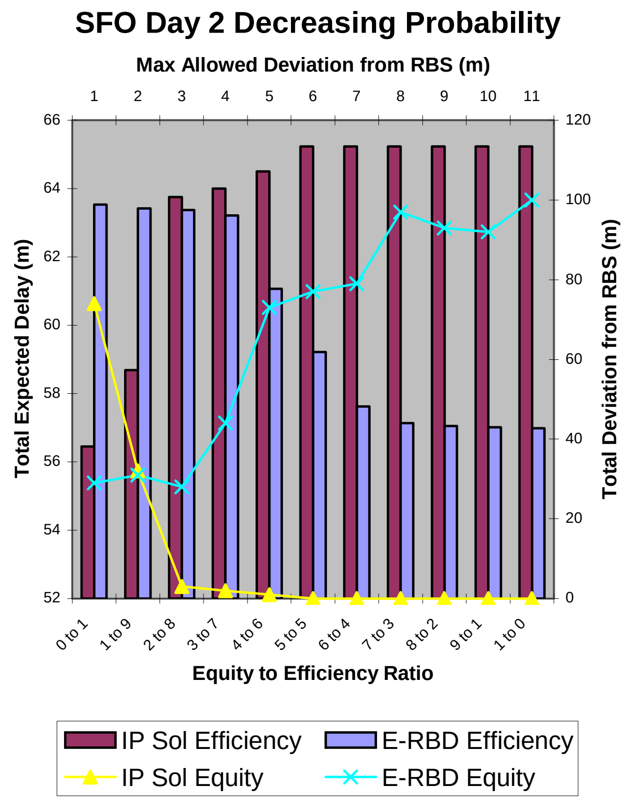


This example provided us with a difference between the efficiency of the RBS and RBD solutions of about 50 minutes. This was the largest difference amongst the examples we considered. Once again we notice the solution provided by the IP with an equity coefficient of 0.1 provides a solution that is efficient (here, within 3 minutes of the total expected delay of the RBD solution), and more equitable than any of the E-RBD solutions. There are also more solutions found in this example by the IP that are not RBD or RBS.

The E-RBD solution, however did not offer such a diverse set of solutions. There were two such solutions, found when delta = 10 and 20. Both these solutions, though, have comparable efficiency to the IP solution with an equity coefficient of 0.1. Even with such a low coefficient though, the IP solution is more equitable than both these E-RBD solutions.

Of the nine test cases considered, each one provided exactly four distinct objective function values for the E-RBD solutions – when δ ranged from 0 to 30. In each case, the solutions when $\delta \geq 30$ have an objective function value equal to the RBD solution. As stated earlier, when $\delta = 0$, this algorithm gives the RBS solution. When $\delta = 30$ we received the RBD solution in each case. So the E-RBD algorithm only provided two non-extreme solutions to compare and contrast with those given by the IP formulation.

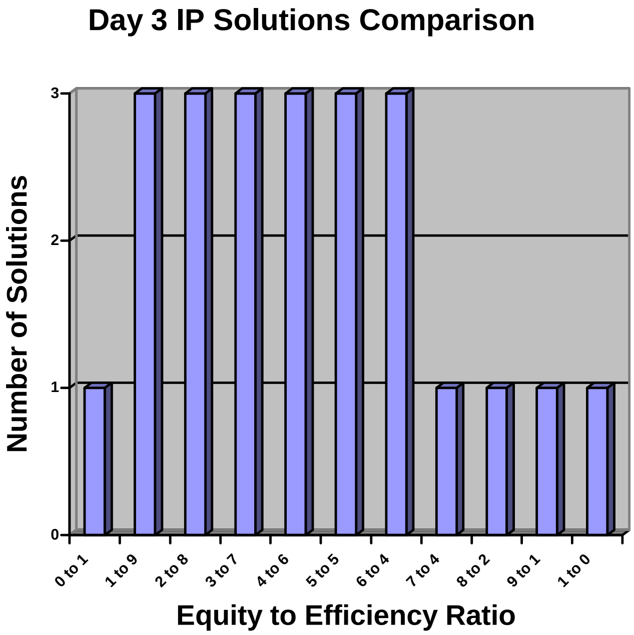
In light of this, we decided to run the E-RBD algorithm with smaller values of δ to see how these solutions compared to the IP solutions. The following graph compares these results to similar results given by the IP, using similar axes. Notice though the change in the δ values.



What we notice in this example is that the E-RBD algorithm begins to give solutions whose total expected delay savings is significant from the RBS solution at $\delta = 5$. This is at a cost of a sharp increase in the total deviation from RBS and the inequity remains high for all remaining $\delta > 5$. It is of corse worth questioning if E-RBD behaves this way in general. Because E-RBD does not take total deviation from RBS into account, it is perhaps not surprising that it does not do as well in finding solutions with low values of this objective.

We were also interested in the role that the probability distribution of the weather clearance time plays on the optimal solutions to the IP. Both RBS and RBD allocate flights to slots without any regard to this probability distribution. The IP is based on a two-stage stochastic IP and thus fundamentally based on this probability distribution.

To test whether the distributions generated different optimal solutions, we used one objective function and output the objective function value of the solutions generated by each probability distribution. Because the probability distribution does not affect the feasibility of solutions to our IP, evaluating these solutions under the same objective function will imply that different solutions with the same objective function value are both optimal. We are thus interested in where the objective function values differ. Below is a graph of the number of feasible solutions and how it corresponds to each ratio.



V. CONCLUSIONS AND FUTURE WORK

We were able to use the IP formulation to generate solutions which have a comparable amount of total expected delay to the E-RBD solution, but which are more equitable. The formulation also gives solutions that are comparable in both equity and efficiency to other rationing principles in the literature such as RBD and RBS.

One thing we noticed here is that both the range of δ considered here and the coefficients for the equity to efficiency ratio were of a very general variety here. It would be interesting to see how these solutions compare with values inside a specified range, say $\delta \in (0, 30)$ and equity coefficients between 0.1 and 0.2. We suspect that the results for E-RBD would be of little difference because the algorithm does not take the total equity of the GDP into consideration when considering allocations. However, it would be of interest to know how close we can have our IP solution to RBD in terms of efficiency, while still remaining within a certain deviation from the RBS solution.

In many instances, though, we also receive an IP solution, where stage 1 is deemed inequitable by E-RBD because too much of the delay is given to one flight or a set of flights. This means that these solutions are not feasible to the more restrictive E-RBD constraints. The only difference in constraints, though, is the limitation placed on flights by δ. This implies that the IP solution may violate this δ constraint for some flights, even when it has low total deviation from RBS.

An additional area to consider, then, is placing equity limitations on each flight in this IP, while still measuring the objective function as a linear combination of equity and efficiency. A benefit of this would be that we would be able to ensure that no single flight or set of flights receives any bulk of the delay, which was a motivating factor behind the development of the E-RBD algorithm, while at the same time minimizing total expected delay and total deviation from the RBS allocation.

We are also currently investigating other properties of this model such as its polyhedron structure and if there is a general form for the optimal solution with similar objective functions. There are also a host of other problems which have a similar structure to this problem, so gaining more understanding of this IP can prove helpful to better understanding those problems as well.

ACKNOWLEGEMENTS

The authors gratefully acknowledge the support of the National Aeronautics and Space Administration Airspace Systems Program under ARMD NRA: NNH06ZNH001

REFERENCES

[1] Ball, M.O., C. Barnhart, G. Nemhauser and A. Odoni, 2007, "Air Transportation: Irregular Operations and Control", *Handbook of Operations Research and Management Science: Transportation,* C. Barnhart and G. Laporte, eds, Elsevier, Amsterdam, pp 1-67.

[2] Ball, M.O., R. Hoffman, and A. Mukherjee. 2008. "Ration-By-Distance with Equity Guarantees: A New Approach to Ground Delay Program Planning and Control", to be published in Transportation Science.

[3] Ball, M.O., R. Hoffman, A. Odoni, and R. Rifkin. 2003. "A Stochastic Integer Program with Dual Network Structure and its Application to the Ground-Holding Problem", Operations Research, 51, 167-171.

[4] Bertsimas, D., and S, Stock Patterson. 2000. "The Traffic Flow Management Rerouting Problem in Air Traffic Control: A Dynamic Network Flow Approach", Transportation Science, 34, 239-255.

[5] Kotnyek, B., and O. Richetta. 2006. "Equitable Models for the Stochastic Ground-Holding Problem Under Collaborative Decision Making", Transportation Science, 40, 133-146.

[6] Manley, B., Sherry, L., 2008, "The Impact of Ground Delay Program (GDP) Rationing Rules on Passenger and Airline Equity", ICRAT.

[7] Mukherjee, A., and Hansen, M., 2007 "A Dynamic Stochastic Model for the Single Airport Ground Holding Problem", Transportation Science, 41, 444-456.

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

[9] Vossen, T., M.O. Ball, R. Hoffman, and M.Wambsganss. 2003. "A General Approach to Equity in Traffic Flow Management and its Application to Mitigating Exemption Bias in Ground Delay Programs", Air Traffic Control Quarterly, 11, 277-292. (a preprint was published in Proceedings of 5th USA/Europe Air Traffic Management R&D Seminar, 2003).