Algorithms for Dynamic Resequencing of En Route Flights to Relieve Terminal Congestion

James C. Jones and David J. Lovell
Department of Civil and Environmental Engineering
University of Maryland
College Park, MD, U.S.A
jonesjc1@umd.edu, lovell@umd.edu

Michael O. Ball
R. H. Smith School of Business
University of Maryland
College Park, MD, U.S.A
mball@rhsmith.umd.edu

Abstract— **In this paper, we present four algorithms designed to reduce the amount of delay in the terminal phase of flight. These algorithms dynamically resequence flight arrival times of en route flights to both transfer and eliminate delay. We propose a strategy for transferring data between the systems and command centers and develop a process for assigning Controlled Times of Arrivals to flights en route. Each algorithm is designed to prioritize criteria of fuel savings, throughput and equity but assigns different weights to these criteria. The first three algorithms are variants of the Ration-by-Schedule algorithm, while the last uses an integer program to assign arrival times. A set of data generated from an ADL file was used to evaluate our algorithms. Analysis suggests that RBS-based algorithms provide strong throughput performance. This improvement in throughput is likely achieved, however, at the expense of fuel usage.**

I. INTRODUCTION

In recent years rising fuel prices and a growing concern for the environmental impact of aviation coupled with a need to accommodate future growth in air travel has motivated increased interest in the development of efficient Air Traffic Management (ATM) practices. In particular, the benefits of Continuous Descent Arrivals/Optimized Profile Descents (CDA/OPD) have been demonstrated over a number of studies [1] [2] [3] [4]. These studies suggest that the maneuver has the potential to yield considerable fuel and emissions savings and reduce noise levels in the surrounding airport communities. Despite its promise, however, widespread implementation of CDA has been marred by safety concerns. Due to the large number of potential flight conflicts near the terminal, the maneuver imposes heavy challenges on existing ATM resources in the presence of heavy congestion.

Fluctuations in weather impose additional challenges to implementation. In the current environment, changes in weather patterns can force air traffic managers to delay flights both on the ground and in the air. In the former case ground delay programs are deployed, and delayed flights are often heavily motivated to rush to the terminal to minimize the total amount of flight delay incurred. In the latter case flights are often vectored to temporarily reduce the flow of traffic into the terminal. In both cases the influx of additional flights imposes a considerable burden on air traffic controllers. This burden can result in considerable flight delays inside the terminal. The presence of terminal congestion has proven itself a considerable obstacle toward the implementation of CDA.

Previous studies have sought to examine the benefits of implementing CDA-type procedures in the presence of congestion. Cao et al. [5] proposed a means of achieving conflict-free CDA in a heavily congested environment and examined the trade-offs between fuel, throughput and time savings with CDA implementation. This study proposed a heuristic that used an integer program (IP) and a bubble sorting algorithm to optimize throughput while achieving the minimum in-trail separations near the terminal. While this study showed that substantial savings could be achieved through intelligent ATM, the proposed algorithm assumes static information over the time period of interest and focuses exclusively on delays as a means of achieving the desired spacing requirements.

Other means of ATM management have also been proposed to alleviate the congestion imposed by terminal and en route traffic. The Terminal Area Precision Scheduling and Spacing System (TAPSS) system builds upon the FAA's Traffic Management Advisor (TMA) system [6]. This system enhances strategic and tactical planning though improved route prediction and constraint scheduling to allow air traffic controllers to optimize capacity and accommodate more fuel efficient maneuvers inside the terminal. The Airline Based En Route Sequencing and Spacing tool sends speed advisories to the Airline Operations Centers (AOC)s to allow crews to more actively manage their speeds in the en route phase of flight [7]. These systems could prove useful for trajectory and speed management; however, simpler alternatives may be possible through the assignment of Controlled Times of Arrival (CTAs).

Other studies have examined the effect of transferring the delay from the terminal to other phases of flight. Knorr et al. [8] proposed a method for calculating the benefit of CDA implementation on flights 90 minutes from the runway and estimated the fuel and emissions savings achieved by ATM controlled speed reductions during cruise. Their estimates suggest that considerable savings can be achieved through controlled speed reductions during the en-route phase of flights. The work illustrates an opportunity for greater systems level management during the cruise phase of flight but did not describe any procedure for ATM-managed speed control.

This paper examines the benefits of assigning dynamically allocated CTAs to transfer terminal delay to the en-route portions of flight. In section 2 we outline a process for transferring relevant data between existing systems to inform all relevant parties. In section 3 we formulate 3 variations of a greedy slot algorithm based on the Ration-by-Schedule (RBS) principles. We also propose an additional algorithm that incorporates an IP to optimize allocation of assignment times. Both algorithms dynamically resolve the problem over a number of time periods. Section 4 provides a case study in which we evaluate and compare the performance of our algorithms under normal and inclement weather conditions at Atlanta Hartsfield-Jackson airport.

II. METHOODLOGY

A. System Description

A scheme for transferring data between the relevant stakeholders over the course of a flight can be readily implemented using existing systems. The traffic flow management system (TFMS) integrates real time flight and weather data and can be used to provide estimated times of arrival (ETAs), scheduled times of arrival (STAs), landing times and flight tracking data. It receives periodic weather updates from the environmental research laboratory (ERL) at the National Oceanographic and Atmospheric Administration (NOAA). It also receives flight plans and cancellation information from the Airline Operational Control Centers (AOCs).

Given the availability of existing technology we see two primary options for assigning CTAs. In the first, CTAs are assigned directly by the Air Route Traffic Control Center (ARTCC) via a voice communication link. Pilots would receive a specified CTA and waypoint from air traffic controllers and would be expected to adjust their speed accordingly to reach the waypoint at the assigned CTA. In the second scheme CTAs would be assigned by the Air Traffic Control Systems Command Center (ATCSCC). The AOC would then relay the CTA to the aircraft through the Aircraft Communications Addressing and Reporting System (ACARS). This implementation could allow some form of brief negotiation between the ATCSCC and AOC to provide better assignment, and possibly to include Collaborative Decision Making (CDM)-type features.

Both approaches have their strengths and weaknesses. In the first scheme, CTA assignments might be taken more seriously when issued directly from the air traffic controllers because there is direct oversight. Since CTA decisions are issued alongside other decisions in real time, the approach might provide better alignment with other controller directives. The approach comes at a cost, however, as it places an additional burden on ATC staff and increases the training needs of centers. The second approach can be implemented quickly as it adds little responsibility to the air traffic controllers' duties. Moreover, since the AOC is directly involved in the process, the scheme could eventually be modified to include some form of collaborative decision making. Since assignments are issued through the AOCs there is some question as to whether airlines could find ways to game the system to the detriment of their competitors and it is important that such issues be examined prior to any collaborative approach being implemented. Diagrams illustrating the flow of information between parties in both schemes are shown in Figures 1a and 1b.

In the longer term, this information would no doubt be provided via datalink. However, our goal is to devise an approach that could be implemented within the new few years.

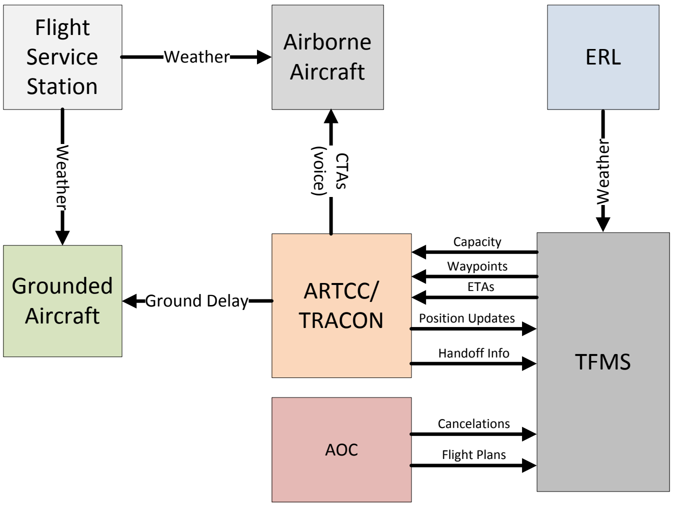

Figure 1a: Data flow between systems and command centers under a centralized control implementation.

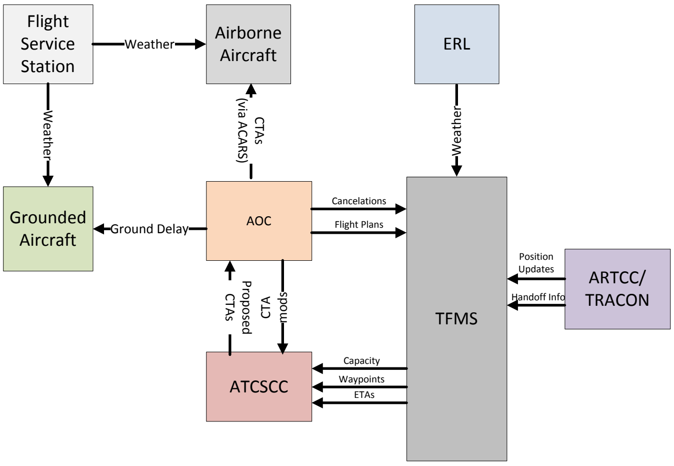

Figure 1b: Data flow between systems and command centers under a collaborative implementation.

B. Process Description

Under the envisioned process, flights are assigned a CTA once they reach a distance of 500 nmi from the destination airport. This paper examines performance criteria and algorithmic approaches for assigning these CTAs. In order to effectively model the CTA assignment problem, estimated flight times are continuously updated as new information is received from the TFMS. Once flights leave the airport they pursue a given route towards the destination airport. As a flight gets closer to the destination, the information on its ETA becomes increasingly reliable. The ETAs provide a forecast of

---

This work was supported by NEXTOR-II, under a Contract with the Federal Aviation Administration; sponsor agreement number: DTFAWA11D00017. Any opinions expressed herein do not necessarily reflect those of the FAA or the U.S. Department of Transportation.

the degree of congestion and the resultant excess flight time and maneuvering in the terminal area. The assignment of CTAs effectively adjusts the ETAs to provide a more orderly flow of traffic into the terminal area. To model the CTA assignment problem we define a set of airport arrival slots that are consistent with the existing airport acceptance rate. The various algorithms we propose assign flights to an arrival slot. The slot time becomes that flight's CTA. To provide a longer-term view of traffic demand, arrival information from flights at distances of up to 1000 nmi from the airport are incorporated into the scheduling algorithms, even though they are not issued a formal assignment until they reach the 500 nmi boundary.

The ANSP updates the list of flights available for scheduling every 15-30 minutes. As these flights approach a distance of 500 nmi, the ANSP sets the number of slots that can be allocated each hour based on the capacity of the airport. The CTA-assignment algorithm then assigns a CTA to all unassigned flights that are approaching the 500 nmi boundary. The flights closest to the boundary, e.g. those that will cross the boundary within 15 minutes, are given their CTAs using one of the communications mechanisms discussed previously. Each CTA is then entered into the aircraft flight management system (FMS). The process will then iterate. Specifically, the next group of flights approaching the 500 nmi boundary is considered and a set of CTAs assigned. Note that there will generally be overlap between the set of flights considered from one iteration to the next as only the closest-in flights are given the computed CTAs. Thus, the CTAs computed for the further-out flights are temporary; these flights are included to provide an assignment procedure with a more global perspective of total flight demand.

III. DESCRIPTION OF ALGORITHMS

There are three key performance criteria we consider in modeling this problem;

1) Fuel usage and delay transfer: the principle motivation for the overall procedure is to allow efficient and unimpeded trajectories in the terminal area. To accomplish this, we seek to transfer delay from the terminal area to the en route portion of the flight, where it can be more efficeintly absorbed.

2) Arrival throughput: it is important to maintain a high throughput into the airport, while accomplishing 1). In fact, ideally the system will increase arrival throughput.

3) Equity: in assigning ETAs, it is invitable that the natural order of the arriving flights will be peturbed. It is important that any flight prioritization be carried out in an equitable manner.

These three performance criteria may be embedded in different algorithmic approaches in different ways. For example, we start out by defining a set of slots consistent with the existing airport arrival rate. By enforcing the restriction that each flight is assigned to some slot we implicitly address criterion 1). For example, if a flight is assigned a slot / CTA that is, say, 10 minutes later than its ETA, that flight will be forced to slow down, and absorb delay during the en route portion of the trajectory. The first three algorithmic approaches we present employ variants of the Ration-by-Schedule (RBS) algorithm. RBS was developed in conjunction with Ground Delay Programs (GDPs) using collaborative decision making (CDM) principles and has become an accepted standard for equitable resource allocation (criterion 3)) (see Vossen and Ball, 2006) [9]. There are a number of important scheduling limitations with implementing the algorithm on airborne flights. While flights can be delayed on the ground for long stretches of time, we cannot impose the same delay lengths in the air due to fuel limitations. Moreover, issuing large airborne delays through vectoring imposes a considerable burden on air traffic controllers. In addition it is often advantageous from a system perspective to speed up flights when other flights are immediately behind them as it reduces the size of the arrival queue. These speed-ups do not always yield universal benefits as they can add to the fuel costs of the flights being sped up. We have tried to consider all of these factors in our algorithm development.

We developed 4 algorithms to prioritize flight assignments. Each algorithm dynamically assigns CTAs over a two-period rolling horizon. The first three use a variant of RBS which assigns each flight to the first ``feasible" slot available. The definition of feasible changes depending on the certain assumptions as discussed below. The last algorithm solves an integer program to assign CTAs based on a dynamic cost function that models the multiple performance criteria mentioned earlier.

A. Greedy Slot Scheduling

Greedy slot algorithms work by dynamically scheduling flights to the first available slot. The algorithms use the ETAs from the two upcoming time periods to allocate slots to flights with ETAs in the first time period. By doing this we help to ensure that scheduling decisions made in the most recent time period have minimal adverse impact on future periods. A description of the algorithm is shown below.

<u>Greedy Slot Assignment</u>

- Let T_j coincide with the end of time period j
- Let $F_1=\{f_{11},\ldots,f_{n1}\}$ be the set of flights with t-ETA< T_1
- Let $F_2=\{f_{12},\ldots,f_{m2}\}$ be the set of flights with $T_1<t$-ETA< T_2
- Let $F_3=\{f_{13},\ldots,f_{p3}\}$ be the set of flights with $T_2<t$-ETA<T_3
- Let $\Omega = \bigcup_{i=2}^{3} F_i$

Step 1: Order flights in Ω by increasing scheduled time of arrival

Step 2: Select the first flight in Ω that has not been assigned to a slot

- If all flights in Ω have been assigned, wait for the next update and repeat step 1
- Otherwise, assign the first flight to the earliest possible unassigned slot

1) Delay Assignment

The GS-delay algorithm seeks to transfer delay from the terminal to the en route portion of flight by assigning a delay to each flight. Flights are assigned the first available slot based on RBS prioritization. This algorithm has the property of minimizing the assigned delay to flights that are running the furthest behind schedule. Thus flights that are delayed due to factors such as mechanical difficulties or runway delays at takeoff are minimally burdened with additional delay.

2) Assignment with Speed-ups:

The previous algorithm attempts to transfer delay; however it does nothing to eliminate it. Time windows can also be left completely open prior to a group of flights that arrive in bulk. When this occurs there are opportunities to eliminate portions of terminal delay by moving up flight arrival times. The GS-speed-up algorithm makes use of both delays and speed-ups to both transfer and eliminate terminal delay. RBS is used to assign flights to the first available slot less than 5 minutes prior to its original ETA.

3) Incorporating Airline preferences:

While speeding up certain flights might yield considerable throughput gains by eliminating existing terminal delay, it does impose a cost on flights that are forced to travel faster than their desired speed. Under these conditions it is not entirely clear that they should be forced to pay this cost. In the previous two algorithms, however, there was no mechanism for flights to avoid such CTA assignments. By allowing airlines to opt out of speed-ups on certain flights prior to assignment the GS-preference algorithm helps them to better deal with their reassignment costs.

<u>Assignment with Airline Preferences</u>

Step 1: AOC expresses speed-up preferences on each flight

Step 2: Order flights in Ω by increasing scheduled time of arrival

Step 3: Select the first flight in Ω that has not been assigned to a slot

- If all flights have been assigned, wait for the next update and repeat step 1
- Otherwise, assign the first to the earliest possible unassigned slot

B. Optimizing Social Welfare

While the greedy slot algorithms provide a convenient and straightforward means of scheduling, they yield suboptimal solutions. An alternative to this approach involves minimizing total flight delay while discouraging speed changes which lead to either heavy additional fuel use or large flight delays. We define our variables and parameters as follows:

- T_j - Time period in which CTA are proposed
- F_j - Set of all flights in T_j
- S - Set of all slots
- c_{ij}^{k}- The cost of assigning flight i in period j to slot k
- $x_{ij}^{k} = \begin{cases} 1 \text{ if flight } i \text{ in period } j \text{ is assigned to slot } k \\ 0, \text{ otherwise} \end{cases}$
- $y_k = \begin{cases} 1 \text{ if slot } k \text{ was assigned on a previous iteration} \\ 0, \text{ otherwise} \end{cases}$
- β- The discount factor applied to the future estimates
- t_k –The CTA corresponding to slot k
- t_{oi} –The slot closest to ETA of flight i
- t – Length of time corresponding to one slot

$$\text{minimize} \sum_{i\in F_1, k\in S} c_{i1}^{k} x_{i1}^{k} + \sum_{i\in F_2, k\in S} \beta c_{i2}^{k} x_{i2}^{k} \tag{1}$$

$$\text{S.T.} \qquad \sum_{k\in S} x_{ij}^{k} = 1 \quad \forall i \in F_j, \forall j \in \{1,2\} \tag{2}$$

$$\sum_{i\in F_1} x_{ij}^{k} + \sum_{i\in F_2} x_{ij}^{k} \leq 1 \quad \forall k \in S \tag{3}$$

$$x_{ij}^{k} \leq 1 - y_k \ \forall i \in F_j, \forall k \in S, \forall j \in \{1,2\} \tag{4}$$

$$x_{ij}^{k} \in \{0,1\}, y_k \in \{0,1\} \tag{5}$$

Equation (2) states that in each time period every flight is assigned to one slot. Equation (3) states that each slot can be assigned to no more than one flight. Equation (4) states that if a slot was assigned to a flight on a previous iteration of the algorithm it cannot be reassigned. Our cost coefficients will vary based on the amount of time between their corresponding slot and the ETA. We used a piecewise linear function to account for fuel, equity and throughput costs. This function can be described by the following expression:

$$c_{ij}{}^{k} = \begin{cases} t_{oi} - t_k, t_k \leq t_{oi} & (6) \\ 0.1*(t_k - t_{oi}), 0 \leq t_k - t_{oi} \leq 2t & (7) \\ 0.8*(t_k - t_{oi}) + 1.5(t_k - t_{oi}), t_k - t_{oi} > 2t & (8) \end{cases}$$

Equation (6) imposes a fuel penalty for speed-ups but imposes no penalty on throughput. Equation (7) imposes a small throughput penalty for minor deviations from the ETAs. Recognizing that flights with considerable delays can impose a significant burden on both throughput and ATM, equation (8) imposes a larger penalty for throughput and incorporates a small fuel penalty as well. Under these conditions minor delays and occasional speed-ups are encouraged while major delays are discouraged and large speed-ups are prohibited. This cost function could be modified considerably to accommodate varying priorities.

IV. EXPERIMENT

A computational experiment was developed to compare and evaluate our algorithms. Two scenarios were developed using historical data. In this section we describe the scenarios and associated assumptions, we present our experimental results, and we offer some analysis.

A. Scenario

To develop our scenario we used data collected for January 11, 2011 at Hartsfield-Jackson airport in Atlanta. On the 11th the airport operated under inclement weather conditions and major airlines operated on a reduced schedule. Runway conditions were icy and the capacity was reduced well below normal levels. The data were obtained from an ADL file, which listed flight numbers, collection time, ETA, STA, the origin airport, actual time of departure, runway arrival time, STAR routes and last available fix.

We assumed an AAR of 36 flights per hour. We then used this capacity to size our slot intervals. We tested our algorithms over 4 hours between 5:00-9:00 pm. Slots were preallocated with flights passing through the 500 nmi boundary mapping the ETAs of flights within 1 hour of landing between the hours of 5:00-6:00 to assigned slots. This mapping enabled us to constrain our solution and prevent simultaneous arrival times due to rescheduling from the algorithm. Due to the low number of incoming flights we assumed 30 minute time periods.

Hartsfield-Jackson airport has 4 corner posts at the northeast, northwest, southeast and southwest corners of the airport. Arriving flights commonly fly through these corner post fixes and are sent to 2 runways. To simplify the problem we assumed that all flights travel to one corner post and one runway. While this might influence capacity at the terminal we do not attempt to simulate the dynamics after we assign a CTA. Instead we examine the deviations of our assigned CTAs from the predicted ETAs and illustrate the potential for transferring and eliminating delay from the terminal provided there is sufficient capacity inside the terminal.

B. Results

We performed a number of calculations.to estimate the potential benefits of each algorithm. Terminal delay was assessed by subtracting the ETA at the 500 nmi boundary from the ETA reported immediately prior to landing as shown in equation (9) and averaging over all flights. This delay can be reduced either by transferring it to other phases of flight or by eliminating it through assigning flights to CTAs ahead of their ETAs. The GS-delay algorithm takes advantage of the first opportunity while ignoring the second. As such we can quantify its benefit by estimating the amount of delay it issues and then comparing it to the average terminal delay. We obtained the average transferred delay by taking the difference between the assigned CTAs and the original ETAs at 500 nmi as shown in equation (10). The delay eliminated from the terminal was found by taking the difference between the ETA and CTA at 500 nmi. Figure 3 shows the amount of delay issued alongside the average delay.

The total benefit of the GS-speed-up algorithm is slightly more nuanced. Estimating the system performance improvement requires assessing the effect of both delays and speed ups. To do so we attempted to account for each separately. Equation (11a) was used to calculate the delay transferred away from the terminal with the algorithm. In this expression δ_s assumes a value of 1 when a flight is delayed from its original ETA by the algorithm and 0 otherwise. Equation (12a) shows the average change in arrival time relative to the original ETA. The portion of terminal delay eliminated by the algorithm can be quantified by subtracting this average change in arrival time from the average delay transferred when the GS-delay algorithm is used. The benefits of the GS-preference algorithm can be expressed similarly using equations (11b), (12b), and (13b). The resulting performance profiles of the GS-speed-up and GS-Preference algorithms are shown in Figures 3 and 4.

Figures 2a-c illustrate the effect of each algorithm. In Figure 2a the GS-delay algorithm assigns each flight the earliest available CTA (slot) immediately prior to its ETA. Thus the transferred delay can be calculated by looking at the average of the difference between the assigned and estimated times of arrival. Figure 2b shows the assignment process for the GS-speed up algorithm. The transferred delay from the terminal can be obtained by identifying flights that are assigned delays (in this case F3 and F4), taking the difference between the assigned and estimated arrival times and weighting the answer by the total number of flights processed. Since both the GS-delay and the GS-speed up algorithms seek to assign flights to the earliest available slot, the amount of delay savings gained by using the GS-speed up algorithm can be found by taking the difference between the slot assignments in Figure 2a and 2b. Figure 2c shows the effect of the GS-preference algorithm on flight arrival times. The transferred and eliminated delay can be obtained similarly by calculating the average positive delay and evaluating the resulting slot assignments against the baseline shown in Figure 2a.

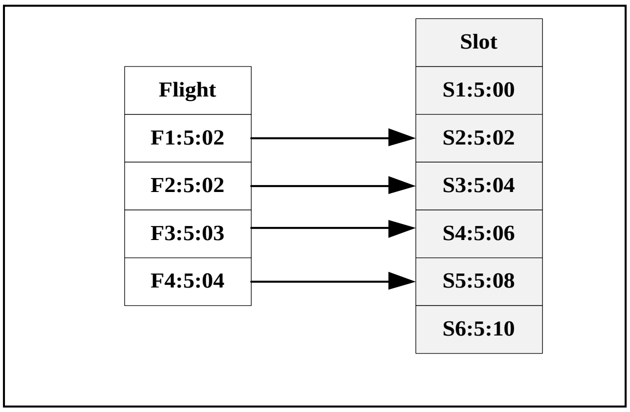

Figure 2a: An example of slot assignment using the GS-delay algorithm.

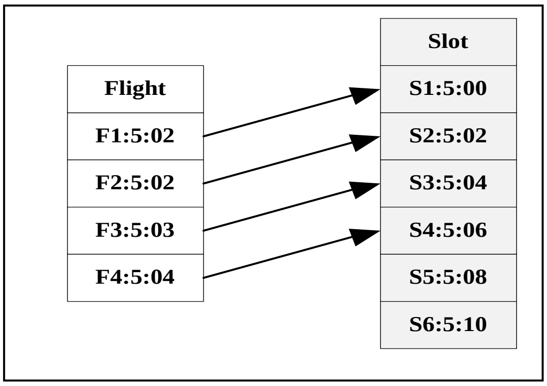

Figure 2b: An example of slot assignment using the GS-speed-up algorithm.

Figure 2c: An example of slot assignment using the GS-Preference.

$$Avg\_D_{\text{Actual}} = \frac{1}{N}\sum_{f\in F}\left(ETA_f^{\,Terminal} - ETA_f^{\,500}\right) \tag{9}$$

$$Avg\_D_{\text{transferred}} = \frac{1}{N}\sum_{f\in F}\left(CTA\_D_f^{\,500} - ETA_f^{\,500}\right) \tag{10}$$

$$Avg\_D_{\text{positive}} = \frac{1}{N}\sum_{f\in F}\left(CTA\_S_f^{\,500} - ETA_f^{\,500}\right)\delta_s \tag{11a}$$

$$Avg\_D_{\text{positive}\_AP} = \frac{1}{N}\sum_{f\in F}\left(CTA\_AP_f^{\,500} - ETA_f^{\,500}\right)\delta_s \tag{11b}$$

$$Avg\_D_{\text{Eliminated}} = \frac{1}{N}\sum_{f\in F}\left(CTA\_S_f^{\,500} - ETA_f^{\,500}\right) \tag{12a}$$

$$Avg\_D_{\text{Eliminated}\_AP} = \frac{1}{N}\sum_{f\in F}\left(CTA\_AP_f^{\,500} - ETA_f^{\,500}\right) \tag{12b}$$

$$Avg\_D_{\text{savings}} = Avg\_D_{\text{transferred}} - Avg\_D_{\text{Eliminated}} \tag{13}$$

$$Avg\_D_{\text{savings}} = Avg\_D_{\text{transferred}} - Avg\_D_{\text{Eliminated}\_AP} \tag{13}$$

The benefits of the IP optimization algorithm can be found using the method for obtaining the benefits of the GS-speed up algorithm described in the previous two paragraphs. Thus a baseline case was used to evaluate the effect of delay savings. In this case we modified the IP to restrict speed-ups and set the discount factor β to 1.2. We then ran another case using our original IP while maintaining our discount factor. The results of our calculations are shown in Figure 6.

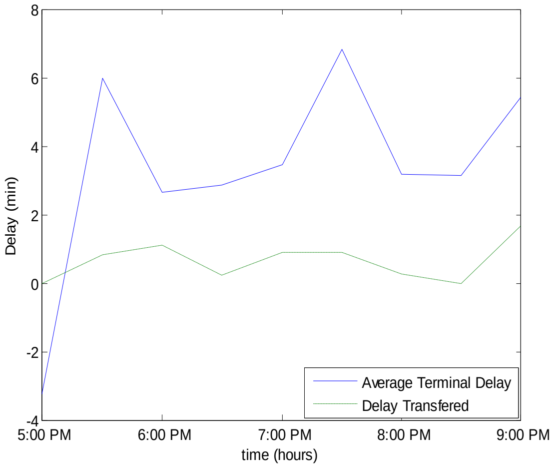

Figure 3: Potential delay transferred en route with the GS-delay algorithm.

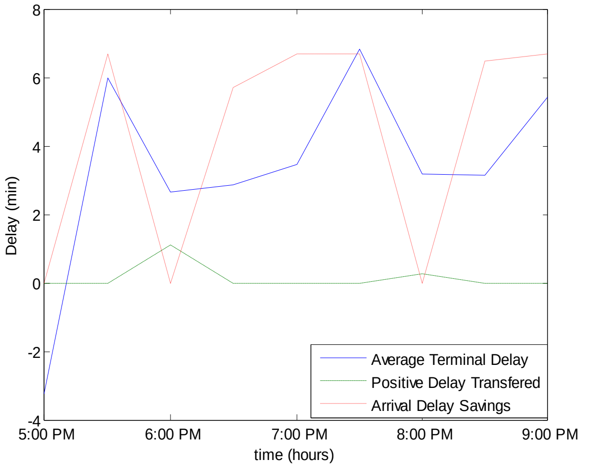

Figure 4: Potential terminal delay transferred and eliminated en route with the GS-speed-up algorithm.

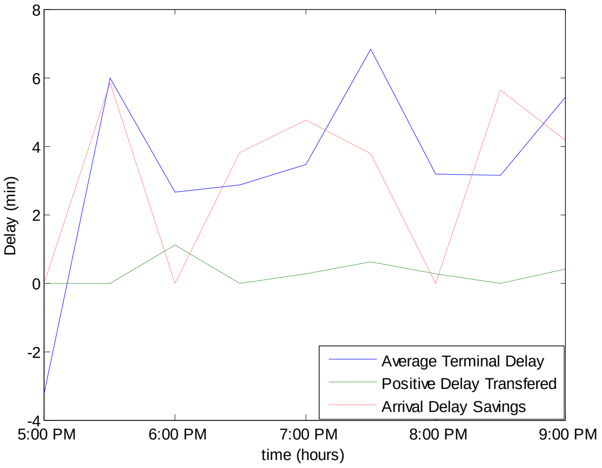

Figure 5: Potential terminal delay transferred and eliminated en route with the GS-Preference algorithm.

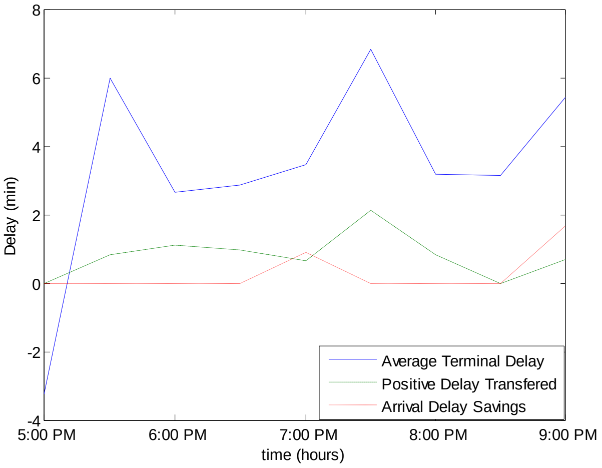

Figure 6: Potential terminal delay transferred and eliminated en route with an IP optimization algorithm.

C. Discussion

Figures 3-6 illustrate the varying degrees of effectiveness of our algorithms. Although all of the algorithms demonstrate some ability to reduce or transfer terminal delay, each addresses our objectives in various ways. Some algorithms prioritize throughput while others give more weight to fuel conservation. A comparison of our algorithms against our performance criteria is shown in Table I. All greedy slot algorithms embody a certain notion of equity in their formulation and received high marks. The GS-preference algorithm imposes certain implementation challenges that are not present in the other algorithms, as it requires airlines to take a more active role in system decision making. As these airlines are in direct competition with one another they might take actions that do not directly benefit the flights they are managing but merely act to make their competitors worse off. These actions could result in worse overall system wide performance, and as such, precautions must be taken to limit this type of behavior. The notion of applying an optimization algorithm to schedule flights is quite common in aviation; however, it lacks the transparency of our greedy algorithms so it was given fair marks.

The experimental results suggest that the GS-speed-up algorithm can be quite effective at eliminating delay. In some instances it even assigns flights to CTAs ahead of their final ETA, thus getting flights to their destination ahead of schedule. It outperforms our optimization algorithm in this respect by a fair margin. This outperformance however, is largely a byproduct of our cost function. As mentioned earlier our objective function attempts to weight criteria of fuel usage, system throughput and equity. These criteria are quite often in conflict with one another. The goals of the GS-speed-up algorithm are not quite so broad. This algorithm seeks to minimize system throughput by assigning flights to the first available slot hastening the arrival time of flights whenever possible. Since the algorithm uses “first scheduled first served” principles it also embodies a certain equity standard. The algorithm does not, however, achieve the same level of fuel savings as our IP algorithm. Although the algorithm assigns delays to flights when prior slots are unavailable, it does not prioritize fuel expenditures to nearly the same extent. As a result it is able to achieve better system throughput performance. While the GS-preference algorithm does not explicitly prioritize fuel savings, either, it does assume that airlines will factor fuel use into their expression of flight preferences. Thus it offers a compromise between our criteria when airlines preferences are heavily tied to fuel use and on-time performance.

It is worth noting that GS-speed, GS-preference and IP optimization algorithms all allow overtakings. As such these approaches will likely add to controller workload prior to 200 nmi where flights are managed with the TMA. We believe that the cost of this marginal increase is offset by the resulting gains in system predictability. By limiting the variation in arrival time we provide air traffic controllers with an added measure of preplanning capability, thereby reducing the amount of ad hoc decision making needed in the airspace near the terminal. A table could be used to measure the equity of the assignments to provide greater system transparency.

TABLE I: A comparison of algorithms against objective criteria

Algorithm	**Fuel Savings**	**Throughput**	**Equity**	**Ease of Implementation**
GS-Delay	High	Fair	High	High
GS-Preference	Fair	High	High	Low
GS-speed-up	Low	High	High	Fair
IP Optimization	Fair	High	Fair	Fair

V. PRELIMINARY CONCLUSIONS AND FUTURE WORK

In this paper we have outlined several methods for reducing terminal delay. We developed an approach for transferring information between relevant systems and command centers and outlined a process for implementing CTA assignment during the en route phase of flight. We then proposed 4 algorithms for dynamic resequencing of flights designed to transfer delay away from the terminal to the en route phase of flight and/or completely eliminate portions of terminal delay. A performance analysis using ADL data showed that the GS-speed-up algorithm had strong potential to transfer and eliminate delay. Our IP optimization algorithm performed reasonably well; however it did not achieve the same delay savings as our GS-speed up algorithm. This is largely an artifact of the cost function and could be improved upon by assigning greater priority to throughput in our cost function. The GS-preference algorithm showed promise and might serve as a good compromise as it allows the airlines to have a greater role in system management.

We have demonstrated that CTA assignments can be used to transfer a considerable proportion of the delay from the terminal to the en route portion of flight. Future work will be oriented towards developing a simulation to test our algorithms and we are currently engaged in those efforts. Additional work to add greater realism to our assumptions about AARs could also be pursued. In reality the actual AAR rarely matches the planned AAR exactly. To evaluate the impact of this phenomenon we will need to incorporate stochastic variation in AAR at the terminal into our model.

Although our RBS-based algorithms do not reward noncompliers due to limitations on fuel and ATM resources, we cannot delay flights indefinitely. Thus the present approach does not currently have a comprehensive mechanism to enforce compliance. We are currently developing a table to track penalized airlines and noncomplying flights. This information could be used to provide additional disincentives for noncompliance. For example airlines that repetitively arrive ahead of their assigned CTA could be penalized on future flights by assigning them lower priority in CTA scheduling. Similarly airlines who have been disproportionately penalized over a sustained time period could be awarded higher priority in future scheduling. By supplementing our current approach with these measures we hope to provide more comprehensive enforcement of our algorithmic objectives.

REFERENCES

[1] Shresta, S., Neskovic, D., and Williams, S. S., "Analysis of Continuous Descent Benefits and Impacts During Daytime Operations," *8th USA/Europe Air Traffic Management R&D Seminar*, Napa, CA, June 29-July 2, 2009.

[2] Robinson, John E, III and Kamgarpour, Maryam, "Benefits of Continuous Descent Operations in High-Density Terminal Airspace Under Scheduling Constraints", *American Institute of Aeronautics and Astronautics*, Anaheim, CA, 2010.

[3] J. K. Klooster, A. Del Amo, and P. Manzi, "Controlled Time-of-Arrival Fight Spacing Analysis," in *9th USA/Europe Air Traffic Management R&D Seminar*, Berlin, Germany June 2011.

[4] Clarke, J.-P. B., Ho, N. T., Ren, L., Brown, J. A., Elmer, K. R., Tong, K.-O., and Wat, J. K., "Continuous Descent Approach: Design and Flight Test for Louisville International Airport," *Journal of Aircraft*, vol. 41, No. 5, 2004, pp. 1054-1066.

[5] Cao, Y., T. Kotegawa, D. Sun, D. DeLaurentis and J. Post, "Evaluation of Continuous Descent Approach as a Standard Terminal Airspace Operation, ," in *9th USA/Europe Air Traffic Management R&D Seminar*, Berlin, Germany June 2011.

[6] Moertl, Peter, M. "Airline Based En Route Sequencing and Spacing Fild Test Results: Observations and Lessons Learned for Extend Metering," in *9th USA/Europe Air Traffic Management R&D Seminar*, Berlin, Germany June 2011.

[7] Swenson, Harry, N., Thipphavong, Jane,Sadovsky, Alex, Chen, Lian, Sullivan, Chris, Martin, Lynne, "Desing and Evaluation of the Terminal Area Precision Scheduling and Spacing System," ," in *9th USA/Europe Air Traffic Management R&D Seminar*, Berlin, Germany June 2011.

[8] Knorr, Dave, Chen, Xing, Rose, Marc, Gulding, John, Enaud, Phillippe, and Hegendoerfer, Holger, "Estimating ATM Efficiency Pools in the Descent Phase of Flight: Potential Saving in both Time and Fuel," *in 9th USA/Europe Air Traffic Management R&D Seminar*, Berlin, Germany June 2011.

[9] Vossen, T. and M. Ball, "Optimization and Mediated Bartering Models for Ground Delay Programs", *Naval Research Logistics*, vol 53, 2006, pp. 75 – 90.