2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Fuel Cost, Delay and Throughput Tradeoffs in Runway Scheduling

Hanbong Lee and Hamsa Balakrishnan

***Abstract*— A dynamic programming algorithm for determining the minimum cost arrival schedule at an airport, given aircraft-dependent delay costs, is presented. Aircraft arrival schedules are subject to a variety of operational constraints, such as minimum separation requirements for safety, required arrival time-windows, limited deviation from a nominal or FCFS sequence, and precedence constraints on the arrival order. In addition to determining schedules that satisfy these constraints, the proposed approach makes it possible to evaluate tradeoffs in terminal-area schedules, through the comparison of throughput maximizing and delay minimizing schedules. A comprehensive analysis of the tradeoffs between average delay and fuel costs is also conducted, accounting for both the cost of delay and the increased rate of fuel burn incurred by an accelerating aircraft.**

I. Introduction

Air traffic congestion is widely considered to be one of the principal constraints to the future growth of the global air transportation industry [1]. An important step toward successfully meeting the increased demand for air traffic services is increasing the efficiency of arrival and departure operations in the terminal-areas surrounding airports. Maximizing efficiency while maintaining safety in super-density terminal areas is one of the major goals of the Next Generation Air Transportation System (NextGen) [2].

Runway schedules need to satisfy the different operational constraints that are imposed by the system. In addition, there are several possible objectives that may need to be optimized while scheduling runways: from the perspective of air traffic control, throughput and average delay are important metrics, while from the airline perspective, the operating costs, especially fuel costs, are important. It is often necessary to make tradeoffs between these objectives while scheduling aircraft operations.

This paper presents an algorithm for optimizing a sum of aircraft delay costs by extending a framework previously used for optimizing runway throughput and for determining the tradeoff between schedule robustness and throughput [3, 4]. The proposed approach can determine the schedule that minimizes the total delay cost in computation time that scales linearly both in the number of aircraft and the largest difference between the latest and earliest arrival time over all aircraft. The common framework allows us to evaluate the tradeoffs between the average delay and throughput, and fuel costs and average delay, for real-world scenarios based at Dallas Fort Worth International Airport.

Hanbong Lee is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. hanbong@mit.edu

Hamsa Balakrishnan is with the Department of Aeronautics and Astronautics, and Engineering Systems Division, Massachusetts Institute of Technology. hamsa@mit.edu

This research was supported by NASA under the NGATS ATM-Airspace Program.

II. Problem definition

A. Constraints

1) Limited deviation from FCFS: Resequencing aircraft can increase the runway capacity in the terminal area, but it also adds to the workload of air traffic controllers. It might be not possible for controllers to implement an efficient sequence that deviates significantly from the nominal or First-Come-First-Served (FCFS) order. This limited flexibility motivates Constrained Position Shifting (CPS) methods. The CPS method, first proposed by Dear [5], stipulates that an aircraft may be moved up to a specified maximum number of positions from its FCFS order. The CPS method helps maintain fairness among aircraft operators and increases the predictability of landing times [3,4]. We denote the maximum number of position shifts allowed as k ($k \leq 3$ for most runway systems), and the resulting environment as a k-CPS scenario.

2) Arrival time windows: Once an aircraft arrives at the Center boundary, tools such as the Trajectory Synthesizer determine the estimated time of arrival (ETA) at which the aircraft will land on an assigned runway, assuming it follows a nominal route and speed profile [6]. If the aircraft is speeded up, the actual time of arrival will be earlier than the estimated time of arrival. The earliest time of arrival is usually limited to one minute before the ETA because of the resultant fuel expenditure. The latest arrival time is determined either by fuel limitations or by the maximum delay that an aircraft can accept. The earliest and latest arrival times of aircraft i are denoted by $E(i)$ and $L(i)$, respectively.

3) Separation requirements: The Federal Aviation Administration (FAA) regulates the minimum spacing between landing aircraft to avoid the danger of wake turbulence. The FAA classifies aircraft into three weight classes (heavy, large, and small) based on the maximum take-off weight [7]. Depending on the weight classes of the two successive aircraft, the minimum separation times between two landings are determined [1]. The minimum required time between the leading aircraft i and the following aircraft j for arrivals is indicated by $\delta_{i,j}$.

4) Precedence constraints: There could also be precedence constraints imposed on the landing sequence. These are constraints of the form that aircraft i must land before aircraft j, and arise due to overtaking constraints, airline preferences from banking operations or high priority flights. Precedence relations are represented by a matrix $\{p_{ij}\}$ such

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

that element $p_{ij} = 1$ if aircraft i must land before aircraft j, and $p_{ij} = 0$ otherwise.

B. Objective functions

There are several possible objective functions that may have to be optimized while determining arrival runway schedules. An important objective, namely maximizing runway throughput (or equivalently, minimizing the completion time of a sequence of aircraft) was considered in prior work, and a dynamic programming based solution approach was proposed [3]. Another desirable objective is to minimize the sum of delay costs, or equivalently, the sum of landing costs of all aircraft in the schedule, given a general landing cost function, $c_i(t_i)$, which corresponds to the cost of landing aircraft i at time t_i. The delay cost could be given by the delay incurred by an aircraft multiplied by a weighting factor representing its priority. Aircraft priority could be based on factors such as crew schedules, passenger connectivity, turnaround times, gate availability, on-time performance, fuel status and runway preferences [8]. When all the weights are equal, minimizing the weighted sum is analogous to minimizing the total delay, or equivalently, the average delay of the sequence.

C. Problem statement

Integrating our objective and constraints, we can pose the following problem for scheduling arrivals on a runway:
Given n aircraft, earliest and latest arrival times $E(i)$ and $L(i)$ for the i^{th} aircraft, separation requirements $\delta_{i,j}$, precedence matrix $\{p_{ij}\}$, landing costs $c_i(t_i)$ for aircraft i landing at time t_i, and the maximum number of position shifts k, we compute the optimal k-CPS sequence and corresponding landing times (t_i) to minimize the *total landing cost*, that is, the sum of the individual landing costs. Without loss of generality, aircraft can be labeled (1, 2, ... , n) according to their position in the FCFS sequence.

III. Algorithm

In prior work, it has been shown that every feasible k-CPS sequence can be represented as a path in a directed graph whose size is polynomially bounded in n and k [3]. This enables us to solve the problem at hand using dynamic programming, as is demonstrated in this section. We briefly describe the generation of this network and its properties.

A. The CPS network

The CPS network consists of n stages, in addition to a source and a sink. Each stage corresponds to an aircraft position in the final sequence. A node in stage p of the network corresponds to a subsequence of aircraft of length $\min\{2k+1, p\}$, where k is the maximum position shift. For example, $n = 5$ and $k = 1$, the nodes in stages $3, \cdots, 5$ represent all possible subsequences of length $2k + 1 = 3$ ending at that stage, while the stage 1 contains a node for every possible sequence of length 1 ending (and starting) at position 1, and the stage 2 contains a node for every possible sequence of length 2 ending at position 2. The network

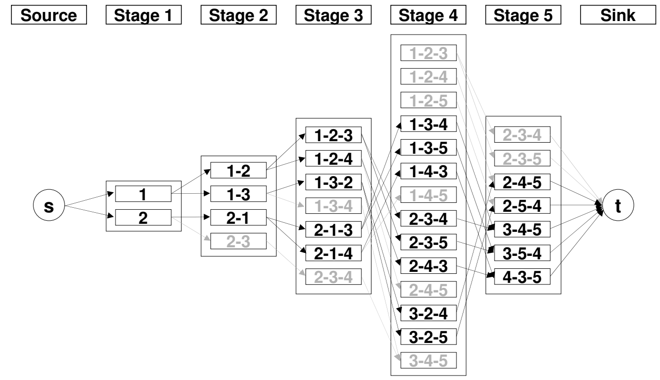

Fig. 1. Network for $n = 5$, $k = 1$.

is generated using all possible aircraft assignments to each position in the sequence.

For each node in stage p, we draw directed arcs to all the nodes in stage $p + 1$ that can follow it. Fig. 1 shows the network for $n = 5$ and $k = 1$. For example, the node (2–1–3) in stage 3 is a successor of node (2–1) in the previous stage (stage 2) and can precede the nodes (1–3–4) or (1–3–5) in the next stage (stage 4). The path (2)→(2–1)→(2–1–3)→(1–3–4)→(3–4–5) represents the sequence (2–1–3–4–5).

Some nodes that violate precedence constraints or are not part of a path from source to sink are removed from the network. These nodes are shown in gray in Fig. 1. By this process, we can produce a "pruned" network, which is significantly smaller than the original network. Precedence constraints further reduce the size of the network.

In the prior work, the following key properties of this network were proved [3]:

(i) Every possible k-CPS subsequence of length $2k+1$ or less is contained in some nodes of the network.
(ii) Every feasible sequence that satisfies maximum position shift constraints and precedence constraints can be represented by a path in the network from a node in stage 1 to a node in stage n.
(iii) Every path in the network from a node in stage 1 to a node in stage n represents a feasible k-CPS sequence.

B. Bounding the makespan

Given a set of arriving aircraft, the *makespan* is defined as the arrival time of the last aircraft or the completion time of the landing sequence. For a fixed set of aircraft (the static case), minimizing the makespan is equivalent to maximizing the throughput of the schedule and is desirable from the perspective of air traffic control. However, the schedule having the minimum makespan is not necessarily the same as the schedule having the minimum average delay [3]. Furthermore, when the cost per unit delay differs significantly between aircraft, the schedule with the minimum total landing cost could be significantly different from the schedule with the minimum makespan. In determining tradeoffs between throughput and landing costs, we would like to determine the minimum landing cost schedule that can be achieved for a given makespan. As a first step, given a

FCFS schedule, we first determine a range of feasible values of the makespan. A trivial lower bound on the makespan is the minimum value among the earliest arrival times of all aircraft that could land last in the sequence. Similarly, the maximum value among the latest arrival times of all aircraft that could land last in the sequence would provide an upper bound on the makespan.

C. Minimizing the total landing cost

For each feasible value of the makespan, we consider all possible k-CPS sequences and determine the optimal schedule that has the minimum total landing cost. This is performed using a dynamic programming recursion. We first define the following variables:

$\ell(x)$: The last aircraft of node x
$\ell'(x)$: The second to last aircraft of node x
$P(x)$: Set of nodes that are predecessors of node x
$\mathcal{I}(j)$: Set of times during which aircraft j could land
$c_j(t)$: Cost of landing aircraft j at time t
t_j: Scheduled time of arrival (STA) of aircraft j

Let $W_x(t_j)$ be the optimal value of sum of the landing costs that is accumulated until the last aircraft of node x lands, given that $l(x)$ lands at time t_j. The objective of this algorithm is to minimize the total landing cost, that is, the sum of landing costs of all aircraft.

$$\text{Total landing cost} = \sum_{i=1}^{n} c_i(t_i)$$

For an arc (x, y) in the CPS network, the sum of landing costs from the first aircraft of the sequence to the last aircraft $\ell(x)$ of node x, $W_x(t_{\ell(x)})$, is used to calculate the sum of landing costs from the first aircraft to the last aircraft $\ell(y)$ of node y, $W_y(t_{\ell(y)})$, using the following dynamic programming recursion:

$$W_y(t_{\ell(y)}) = \min_{x \in P(y)} \left\{ W_x(t_{\ell(x)}) \right\} + c_{\ell(y)}(t_{\ell(y)}),$$
$$\forall\ t_{\ell(y)} \in \mathcal{I}(\ell(y)) : t_{\ell(y)} - t_{\ell(x)} \geq \delta_{\ell(x),\ell(y)}$$

The proof of correctness of this recursion follows standard techniques for proving the validity of dynamic programming recursions and is presented in the appendix for completeness.

For a node y in the first stage, since there are no previous landing costs, its landing cost becomes $W_y(t_i) = c_i(t_i)$, where the last aircraft of the node is i.

Since the state space for $W(\cdot)$ is infinite, the recursion is not computationally practical. In order to implement the algorithm, we discretize all times into periods of length ϵ. Since radar update rates are once every 10-12 seconds [9], it would be reasonable to set ϵ to a value between 1 and 10 seconds.

The dynamic programming recursion presented above determines the landing cost W for all nodes in stage n for all feasible time periods. The minimum landing cost for a given makespan t is the minimum over all nodes x in stage n of $W_x(t_{\ell(x)})$, such that $t_{\ell(x)} = t$. Comparing these minimum landing costs in a range of the feasible makespan, we can determine the sequence and arrival times of aircraft that minimizes the total landing cost.

D. Complexity

It was shown in [3] that the number of nodes in the CPS network is $O(n(2k+1)^{(2k+1)})$ and the number of arcs is $O(n(2k+1)^{(2k+2)})$. In the present case, we have to assign the arrival time of aircraft (and the corresponding landing cost) associated with each arc (x, y) in the given time-window. In a given arc, because all aircraft in the previous node are already assigned completely, the last aircraft of the current node y needs to be assigned to a specific time within the given time-window. When we assume that the length of the largest interval of feasible arrival times among all aircraft is L and the accuracy is ϵ, there are at most (L/ϵ) time-periods in a given arrival time-window. The computational work done per arc in the network is therefore $O(L/\epsilon)$.

Lemma 1: The complexity of the proposed dynamic programming algorithm is $O(n(2k+1)^{(2k+2)}L/\epsilon)$, where n is the number of aircraft, k is the maximum allowed number of position shifts, L is the largest difference between the latest and earliest arrival times over all aircraft, and ϵ is the desired output accuracy.

IV. Evaluating tradeoffs between objectives

A. Minimizing average delay

In the previous section, we presented an algorithm for minimizing the sum of landing costs, given the cost of each aircraft landing at a particular time. The problem of minimizing the sum of arrival times of all aircraft can be solved by setting the cost of landing an aircraft at a particular time to be equal to that time (that is, $c_i(t_i) = t_i$). Since the average delay of a given group of aircraft is equal to the sum of the individual delays (difference between estimated arrival time (ETA) and actual arrival time (STA)) divided by the number of aircraft, we can write

$$\text{Average delay} = \tfrac{1}{n}\left(\sum_{i=1}^{n} STA_i - \sum_{i=1}^{n} ETA_i\right).$$

The sum of estimated arrival times is known and a constant, therefore the problem of minimizing the sum of arrival times is equivalent to that of minimizing the average delay.

We consider a random instance of scheduling a sequence of 30 aircraft on a single runway. Precedence constraints are imposed between aircraft arriving on the same flight jet route, which is assigned to be one of four possible routes. The earliest arrival time is equal to the ETA ($E(i) = ETA_i$), and the maximum allowed delay is one hour. Table I shows the *makespan* (the arrival time of the last aircraft in the group) and the *average delay* for the FCFS and CPS sequences. When the objective is to minimize average delay (columns 2 and 3 in Table I), we note that as the maximum number of position shifts k increases, the average delay decreases. We also compare the schedule that minimizes the average delay with the one that minimizes the makespan (using the CPS framework [3]). For each value of k, for the minimum value of the makespan, we determine the minimum achievable value of the average delay (shown in Table I). We note that the decrease in makespan is achieved at the cost of an increase in the average delay. While it is true that a decrease in makespan frequently results in an improvement in the

Procedure	Min. average delay		Min. makespan	
	Makespan	Average delay	Makespan	Average delay
FCFS	3296	134.27	3296	134.27
1-CPS	3247	101.03	3247	101.03
2-CPS	3247	94.13	3242	103.90
3-CPS	3247	92.30	3232	121.00

TABLE I

COMPARISON OF THE MINIMUM AVERAGE DELAY AND THE MINIMUM MAKESPAN SOLUTIONS.

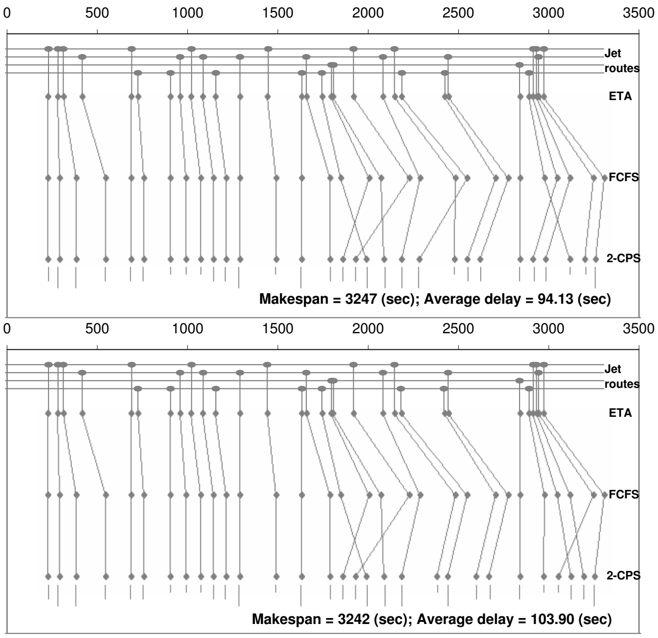

Fig. 2. Simulated arrival traffic for (top) minimum average delay and (bottom) minimum makespan, with 2-CPS. The horizontal axes denote the time line. The ETAs correspond to the estimated time of arrival at the airport if the aircraft flies at its nominal speed and route. The short vertical line beneath each aircraft indicates the type of aircraft: a longer line denotes Heavy, a medium line denotes Large, and a shorter line denotes Small.

average delay [3], this is not necessarily the case. Similarly, minimizing the average delay may result in an increase in makespan. The minimum average delay and minimum makespan schedules for $k = 2$ are shown in Fig. 2.

B. Analysis of tradeoffs between delay and throughput

We further investigate the tradeoff between average delay and throughput that was demonstrated in the previous section using Monte Carlo simulations. We generate 1,000 instances of 30-aircraft sequences, with the aircraft types and jet routes assigned randomly using the appropriate probability distributions [3]. Precedence constraints are imposed among aircraft using the same jet route, and time-windows are assigned with the ETA as the earliest arrival time and a maximum delay of one hour. For each of these generated instances, we optimize the schedule for two different objectives: minimizing the average delay and minimizing the makespan.

The comparison between the two solutions is shown in Fig. 3. The horizontal axis corresponds to the maximum throughput solution and shows its normalized improvement in throughput, $\frac{(CPS\ throughput)-(FCFS\ throughput)}{(FCFS\ throughput)}$. The vertical axis corresponds to the minimum average delay solution and shows its normalized decrease in average delay, which is calculated as $\frac{(FCFS\ avg.\ delay)-(CPS\ avg.\ delay)}{(FCFS\ avg.\ delay)}$.

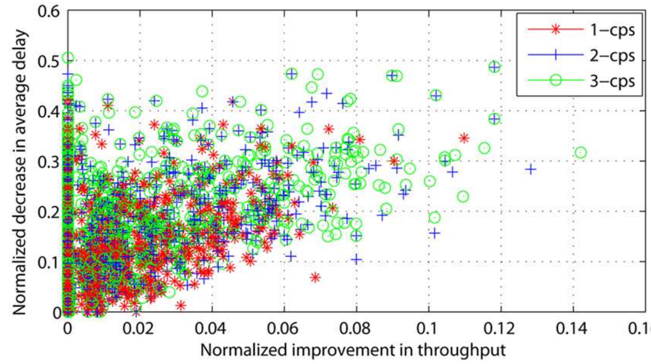

Fig. 3. Normalized runway throughput vs. normalized average delay.

We note that about 45% of the instances in Fig. 3 lie on the vertical axis. This means that in 45% of the instances, there is little or no benefit (over the FCFS schedule) in minimizing the throughput of the sequence, although there are other instances in which up to 14% improvement in throughput can be achieved through resequencing the arrival sequence.

Fig. 4 (left) shows the makespan values of both the minimum makespan schedule and the minimum average delay schedule. We note that the makespan of the schedule that minimizes the average delay does not differ very much from the minimum makespan values. Fig. 4 (right) shows the average delay values of both the minimum makespan and the minimum average delay schedules. While in a large number of instances the average delay values are not much larger than the minimum, there are instances in which the average delay corresponding to the minimum makespan solution is significantly greater than the minimum value that can be achieved. This means that the benefit from maximizing the throughput of the sequence frequently comes at the expense of an increase in the average delay incurred by the aircraft.

We also compare the minimum average delay and maximum throughput schedules to the nominal FCFS schedules. The rationale behind this is as follows: since the minimum average delay solution can have a sub-optimal throughput, it is possible that the throughput of the minimum average delay solution is actually *lower* than the FCFS throughput. Similarly, the average delay of the minimum makespan solution may be higher than the FCFS average delay. Therefore,

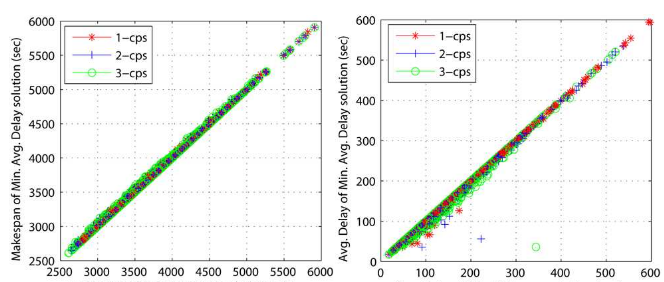

Fig. 4. Comparison between min. average delay schedule and min. makespan schedule in terms of (left) Makespan (right) Average delay.

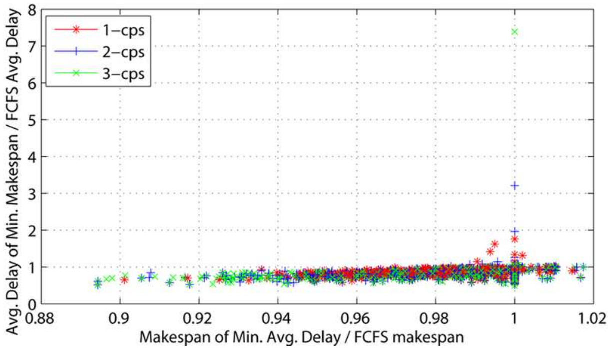

Fig. 5. Comparison between optimal and FCFS schedules.

for the Monte Carlo simulations, we also compare (in Fig. 5) the ratio of the makespan of the minimum average delay schedule to the FCFS makespan and the ratio of the average delay of the minimum makespan schedule to the FCFS average delay.

In most samples, these ratios are less than one, that is, the resequencing using CPS improves both the makespan and the average delay when compared to the FCFS solution. However, some instances have a ratio greater than one, implying a worse throughput or average delay than the FCFS schedule. The results are summarized in Table II. For example, in about 4% of instances, the schedule that minimizes the average delay (with $k = 3$) has a worse throughput than the FCFS schedule. The maximum throughput schedule (with $k = 3$) has a worse average delay than the FCFS schedule in about 5% of instances.

	Min. avg. delay schedule has a larger makespan than FCFS	Min. makespan schedule has larger avg. delay than FCFS
1-CPS	3.6 %	4.0%
2-CPS	4.0 %	4.5%
3-CPS	3.7 %	5.3%

TABLE II

COMPARISON BETWEEN OPTIMAL AND FCFS SCHEDULES.

C. Minimizing fuel costs

The proposed algorithm is applied to the problem of minimizing the fuel costs of the arrival schedule at Dallas/Fort Worth (DFW) International Airport. Fuel costs account for almost 50% of the flight operating costs per block hour for most airlines [10]. Operating costs, including the cost of fuel consumed per unit delay, are dependent on the specific aircraft types and the airlines. In this study, we use the latest fuel costs based on Form 41 data, in which each airline provides the operating cost breakdown (crew costs, fuel costs, insurance, tax, and maintenance costs per block hour of operation) for each aircraft type that it operates [10].

The airport is located in the center of Fort Worth Center airspace (ZFW), as illustrated in Fig. 6. Most arrivals into DFW pass through one of four arrival gates, BYP, CQY, JEN, and UKW, before they enter DFW TRACON airspace. Precedence constraints on the landing sequence are imposed based on aircraft that approach on the same jet route. In this study, we focus on scheduling arrivals onto Runway 18R.

Neuman and Erzberger noted that if an aircraft was allowed to speed up and land before its ETA, it could potentially result in significant savings in delay for the aircraft that follow it [6]. This procedure of allowing the earliest arrival time $E(i)$ to be less than the ETA is known as *Time advance* (TA). However, this decrease in delay (and the associated savings in fuel consumption) is accomplished at the expense of the extra fuel that is consumed in speeding up from the nominal velocity profile.

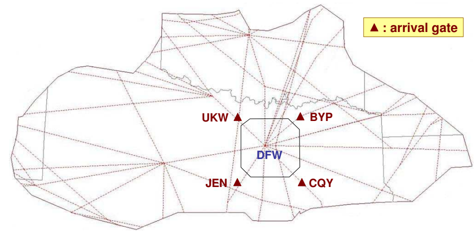

Fig. 6. ZFW airspace, showing jet routes and arrival gates [11].

Using the fuel consumption rates, costs, and elapsed times for both the nominal speed profile and the accelerated profile corresponding to various initial speeds and altitudes [6] and calibrating the fuel costs for the nominal profile with the block hour fuel costs of the American Airlines MD80 aircraft (which account for a significant fraction of operations at DFW), we estimate the cost per minute of time advance for each airline and aircraft type.

The earliest time of arrival is determined by the number of minutes of time advance that is allowed, while the latest time of arrival is chosen such that no aircraft incurs more than 60 minutes of delay. We consider resequencing with the maximum number of position shifts k varying between 1 and 3 and determine the arrival schedule that minimizes the total fuel cost, accounting for both the fuel cost of delay and that of time advance for each aircraft.

The ETAs are assumed to be equal to the arrival times on the airline flight schedules [12]. Aircraft are unable to land at the ETAs in practice primarily because of the minimum separation requirements imposed, in addition to the inability to overtake along a jet route. The FCFS landing sequence therefore produces delay and, as a consequence, additional fuel consumption. The extra fuel costs compared to ETAs for scheduling under CPS are also calculated, and the benefits of the CPS schedule relative to FCFS are evaluated.

We consider intervals of one hour, between 8:00AM and 2:00PM. Fig. 7 shows that when k increases, the fuel cost savings increase, as expected. Similarly, as the allowed speed up increases, the extra fuel cost decreases. However, it is important to note that the marginal benefit is reduced, and the curve seems to level off around a value of 3 minutes time advance. This suggests that over 3 minutes of time advance requires an increase in fuel costs that offsets the fuel-cost benefits of the time advance on the rest of the aircraft. It is interesting to note that while the 12PM-1PM and the 1PM-2PM time-windows have the same number of aircraft ($n = 41$), the form of the plot is quite different, with there being little benefit to time advance of more than a minute in the latter case. A closer look at the schedules for the time-windows shows that while the 12PM-1PM window has 23 precedence constraints, the 1PM-2PM window has

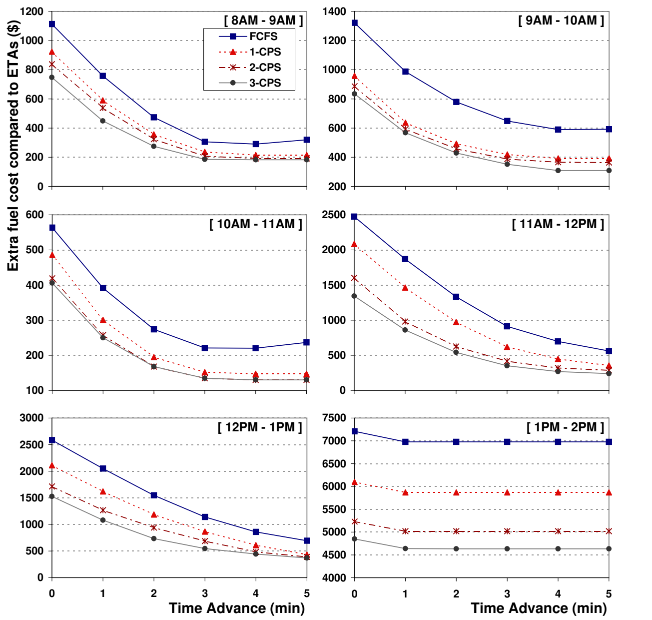

Fig. 7. Extra fuel cost compared to the ETAs vs. the allowed time advance for the minimum fuel cost schedules, for different time-windows.

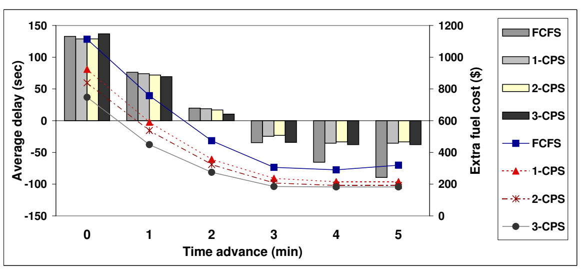

Fig. 8. Average delay (columns) and extra fuel cost (lines) vs. allowed time advance for the time interval 8-9AM.

33 precedence constraints. The heavily constrained sequence prevents aircraft from deriving benefit from time-advance.

We also plot the average delay values for the 8AM-9AM time-window in Fig. 8. This figure shows that the average delay values do decrease as the amount of time advance increases (this was the primary motivation for time advance). However, for a fixed amount of time advance, the decrease in fuel cost may be achieved at the expense of an increase in average delay.

V. Conclusion

This paper presents an algorithm for minimizing the total delay cost of an arrival schedule, in the presence of constraints such as separation criteria, arrival time windows, limits on deviation from the FCFS sequence, and precedence conditions. This algorithm is used to evaluate the tradeoff between the average delay and throughput. The results suggest that significant improvements in the average delay (up to 50%) can be achieved through resequencing with the associated decreases in throughput being reasonably small. We also apply this algorithm to the problem of minimizing a sum of fuel costs, while investigating the policy of allowing aircraft to land earlier than their ETAs (time advance). We determine the minimum fuel cost schedules by considering both the delay cost and the extra fuel cost incurred by speeding up. The analysis suggests that a time advance of up to 3 minutes is advisable in most practical scenarios. Current and future research plans include a more detailed analysis using the total operating costs, as well as extensions to multiple runway scheduling and departure planning.

References

[1] R. de Neufville and A. Odoni, *Airport Systems: Planning, Design and Management*. McGraw-Hill, 2003.

[2] The Joint Planning and Development Office (JPDO), "Concept of Operations for the Next Generation Air Transportation System," Version 2.0, June 13, 2007.

[3] H. Balakrishnan and B. Chandran, "Scheduling aircraft landings under Constrained Position Shifting," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Keystone, CO, 2006.

[4] B. Chandran and H. Balakrishnan, "A dynamic programming algorithm for robust runway scheduling," in *American Control Conference*, New York, NY, July 2007.

[5] R. G. Dear, "The dynamic scheduling of aircraft in the near terminal area," Ph.D. dissertation, Massachusetts Institute of Technology, 1976, flight Transportation Laboratory Report R76-9.

[6] F. Neuman and H. Erzberger, "Analysis of delay reducing and fuel saving sequencing and spacing algorithms for arrival spacing," NASA Technical Memorandum 103880, October, 1991.

[7] FAA, Federal Aviation Administration, "Air Traffic Control: Order 7110.65R. Includes Change 3, February 16, 2006. (Appendix A)," http://www.faa.gov/atpubs.

[8] G. C. Carr, H. Erzberger, and F. Neuman, "Airline arrival prioritization in sequencing and scheduling," in *2nd USA/Europe Air Traffic Management R&D Seminar*, Orlando, FL, December 1998.

[9] H. Erzberger, T. J. Davis, and S. M. Green, "Design of Center-TRACON Automation System," in *AGARD Meeting on Machine Intelligence in Air Traffic Management*, 1993.

[10] Eclat Consulting, "Aircraft operating costs and statistics," *Aviation Daily*, vol. 368, June, 2007.

[11] J. Ott, "Tactical decision aids used by ZFW CWSU," April 2007, www.wrh.noaa.gov/psr/aviation/SAWS_Workshop/25_TDAs_ZFW.ppt.

[12] "DFW airport arrivals," July 2, 2007, http://www.flightstats.com.

Appendix

Proof: We first observe that, by construction, $\ell(x) = \ell'(y)$ for $x \in P(y)$. Therefore, $W_y(t_{\ell'(y)}) = W_y(t_{\ell(x)})$.
Since $W_y(t_{\ell(y)})$ is the minimum value of total landing cost over all paths leading to node y,

$$\begin{aligned} & W_y(t_{\ell(y)}) \le W_x(t_{\ell(x)}) + c_{\ell(y)}(t_{\ell(y)}) \\ & \forall\, x \in P(y), t_{\ell(x)} \in \mathcal{I}(\ell(x)), t_{\ell(y)} \in \mathcal{I}(\ell(y)), \\ & \text{where } t_{\ell(y)} - t_{\ell(x)} \ge \delta_{\ell(x),\ell(y)} \end{aligned}$$

This means that, in particular,

$$\begin{aligned} & W_y(t_{\ell(y)}) \le \min_{x \in P(y)} \{W_x(t_{\ell(x)})\} + c_{\ell(y)}(t_{\ell(y)}) \\ & \forall\, x \in P(y), t_{\ell(x)} \in \mathcal{I}(\ell(x)), t_{\ell(y)} \in \mathcal{I}(\ell(y)), \\ & \text{where } t_{\ell(y)} - t_{\ell(x)} \ge \delta_{\ell(x),\ell(y)} \end{aligned}$$

To complete the proof, we only need to show that the above relationship can never hold as a strict inequality. For contradiction, suppose that

$$\begin{aligned} & W_y(t_{\ell(y)}) < W_x(t_{\ell(x)}) + c_{\ell(y)}(t_{\ell(y)}) \\ & \forall\, x \in P(y), t_{\ell(x)} \in \mathcal{I}(\ell(x)), t_{\ell(y)} \in \mathcal{I}(\ell(y)) \end{aligned}$$

This is equivalent to

$$\begin{aligned} W_y(t_{\ell(y)}) - c_{\ell(y)}(t_{\ell(y)}) \quad &< \quad W_x(t_{\ell(x)}) \\ \forall\, x \in P(y), t_{\ell(x)} \in \mathcal{I}(\ell(x)), &\text{ and } t_{\ell(y)} \in \mathcal{I}(\ell(y)). \end{aligned}$$

$$\begin{aligned} \implies W_y(t_{\ell(y)}) - c_{\ell(y)}(t_{\ell(y)}) \quad &< \quad \min_{\substack{z \in P(y) \\ t_{\ell(z)} \in \mathcal{I}(\ell(z))}} \{W_z(t_{\ell(z)})\} \\ \forall\, t_{\ell(y)} \in \mathcal{I}(\ell(y)) & \end{aligned}$$

However, $W_y(t_{\ell(y)}) - c_{\ell(y)}(t_{\ell(y)})$ is the total landing cost of the subsequence of $W_z(t_{\ell(z)})$ that ends at node z and time $t_{\ell(z)}$. This contradicts the minimality of $W_x(t_{\ell(x)})$ for $x = z$. ■