AN OPTIMIZATION MODEL FOR REDUCING AIRCRAFT TAXI TIMES AT THE DALLAS FORT WORTH INTERNATIONAL AIRPORT

Sivakumar Rathinam*, Justin Montoya, Yoon Jung****
***University of California-SantaCruz, ****NASA Ames Research Center-Moffett Field**

Keywords: *Taxi Scheduling, Surface Operations, Mixed Integer Linear Programming*

Abstract

This paper presents an optimization model for an aircraft taxi-scheduling problem that arises on the surface of airports. Though formulations are already available for this problem in the literature, they either ignore aircraft type or do not include all the safety constraints that are required to keep any two aircraft separated by a minimum distance at any time instant. The proposed model improves on the previously known problem formulation in the literature and formulates the problem as a mixed integer linear program that takes into account the aircraft type and all necessary safety constraints. Solutions produced by the optimization model are then compared with the taxi times produced by a First Come First Serve (FCFS) algorithm to assess the benefits of optimization. Preliminary simulation results demonstrate that approximately six minutes of taxi time per aircraft can be saved for the surface traffic at Dallas-Fort Worth International Airport using the optimization model as compared to the FCFS algorithm.

1 Introduction

As demand and congestion intensifies, reducing aircraft taxi times at airports is becoming increasingly important. There are two main methods to reduce taxi times on airport surfaces. The first method is to hold aircraft at their gates as long as possible and then release them on optimal schedules. The second method is to determine de-conflicted taxi routes and thereby offer more unimpeded paths. Credible solutions for either method must properly account for aircraft type and several safety constraints, which is a challenge. This paper focusses on the first method.

The problem of optimally assigning a time to each aircraft on when to leave its gate, as well as when to reach specific points on the airport surface along its route, is referred to as the taxi scheduling problem in the literature [2],[3],[4]. Though few optimization models are available for the taxi scheduling problem, they either ignore aircraft type [2] or do not account for all the safety constraints [3],[4]. The mixed-integer linear program formulation in this paper accounts for aircraft type and all the key safety constraints. The formulation is also general and can be extended to any airport layout.

To assess the benefits of the optimal solutions, comparisons are made with the taxi times obtained from a first-come-first-served (FCFS) algorithm. A preliminary study was conducted for a group of twenty-five aircraft departing the east side of Dallas-Ft. Worth Airport (DFW). The next section gives a detailed description of the problem along with possible solution approaches. Next, the optimization model is developed, followed by the simulation results and conclusions.

2 Problem setup and background

This paper addresses the following Aircraft Taxi-scheduling Problem (ATP): Given taxi routes for all the aircraft (both arrivals and

departures), the problem is to schedule each aircraft at all the taxi intersections along its route such that the total taxi time of all aircraft is minimized subject to the following constraints:

1. *Safety constraint*: any two aircraft must be separated by a minimum distance at any time instant. Also, any two aircraft must not overtake or cross each other on a taxi link.

2. *Runway occupancy constraint*: each runway must be occupied by only one aircraft at any time instant. Sufficient time must be provided between any two aircraft using a runway depending on whether the aircraft are arrivals or departures.

3. *Origin and destination timing constraint*: each aircraft has to reach its desired destination before a given time. Similarly, a departure aircraft is available at the first intersection or spot[1] in its route only after a given push back time.

4. *Speed constraint*: the maximum speed of an aircraft on a taxi link is bounded by a constant depending on the location of the link.

A taxi route of an aircraft is specified by the sequence of intersections the aircraft needs to travel to reach its desired destination (e.g., runway or gate). Scheduling an aircraft at a given intersection along its route requires specifying the time at which the aircraft must reach that intersection. The collision avoidance constraint also ensures that at any time instant the total number of aircraft on each taxiway link does not exceed the capacity of that link.

The ATP is closely related to the following well known air traffic flow management problem (TFMP) studied by Bertsimas et al. [1]: Given a set of aircraft (with their respective routes), airports, time periods and the pairs of aircraft that follow each other (*i.e.*, if (i, j) is a pair then aircraft j departs from its airport (A) after aircraft i lands at airport A), the problem is to find the release time of each of the departing aircraft, and find the speed of each aircraft in the sectors along its route while satisfying the capacity requirements of the sectors at all time periods. The objective of the TFMP is to minimize the total delay cost of all the aircraft.

Refer to Fig. 1 taken from Ref. [1] showing an example of the routes of two aircraft. A route of an aircraft in the TFMP is specified by the sequence of sectors that the aircraft needs to travel from its departure airport to its arrival airport. The capacity constraint requires that at each time period, there is an upper bound on the number of aircraft occupying any sector. Controlling the speed of an aircraft in a sector along its route is also equivalent to controlling the time the aircraft takes to cross that sector. If one can relate sectors to taxiways, then the TFMP is similar to the taxi scheduling problem addressed in this paper.

However, there is a fundamental difference that separates these two problems. There is no notion of any two aircraft preserving their sequence when they enter and leave a sector in the TFMP. That is, if aircraft i enters a given sector s before aircraft j, aircraft i may leave sector s before or after aircraft j leaves sector s. This constraint is defined as the overtaking constraint in this paper and is not required for the en-route problem considered in [1] as the sector spaces are large. The reader can imagine a sector being a multilane freeway where the minimum separation constraint between any two aircraft does not exist. But this minimum separation constraint is of utmost importance in the ATP (refer to Fig. 2) where a taxiway is essentially a single lane, oneway at any time instant. As a result, the ATP requires the constraint of preserving the sequence of aircraft when they enter and leave a taxiway.

The minimum separation constraint significantly influences the type of model one can choose to formulate the ATP. For example, consider the following choices for formulating the ATP and the respective challenges. One can

[1] A spot is an intersection that is located on the boundary of the ramp area (controlled by the ramp tower controllers) and the taxiways (controlled by the ground controllers)

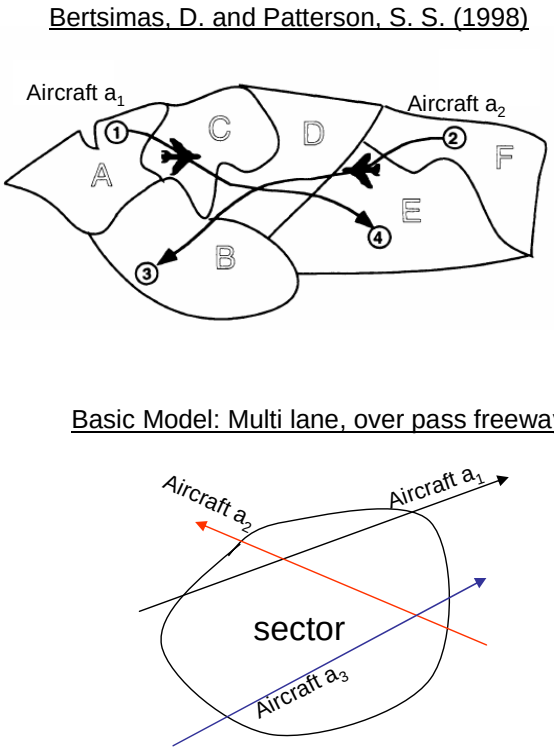

Fig. 1 In the en-route problem addressed by Bertsimas et al. [1], each sector has a capacity constraint without any requirement on the minimum separation between any two aircraft. In the figures, A,B,C,D,E and F denote the sectors. For example, in the top figure, aircraft a_1 travels from airport 1 through sectors C,D to airport 4.

choose to formulate the ATP using the integer program formulated in Bertsimas et al. However, to ensure aircraft do not overtake while traveling along a taxiway, one needs to divide each taxiway into smaller cells so that each cell can at most contain only one aircraft. By modeling this way, one can eliminate the possibility of aircraft overtaking each other inside a cell. Since a cell has to accommodate every type of aircraft, the length of the cell will be determined by the length of the largest aircraft that travels that cell. As a result, apart from increasing the complexity of the problem (increase in the number of variables), dividing a taxi way into smaller cells ignores the type of aircraft.

One can also choose to formulate the ATP using a set of binary variables that track the order in which each of the aircraft visit the intersections and another set of continuous variables that denote the time each aircraft reaches the intersections along its route. In this case, the minimum separation constraints can be formulated as quadratic constraints (refer to subsection 3.2). These quadratic constraints however can be linearized and this leads to a weak Mixed Integer Linear Program (MILP). By a weak formulation, we mean that the optimal LP relaxation cost corresponding to the MILP would not be sufficiently close to the optimal cost of the MILP. If one decides to solve this problem using a Branch and Bound method, then formulating the problem this way might increase the computational time. But there is a distinct advantage in formulating the problem this way because the type of the aircraft can be taken into account.

Problem formulations for the taxi-scheduling

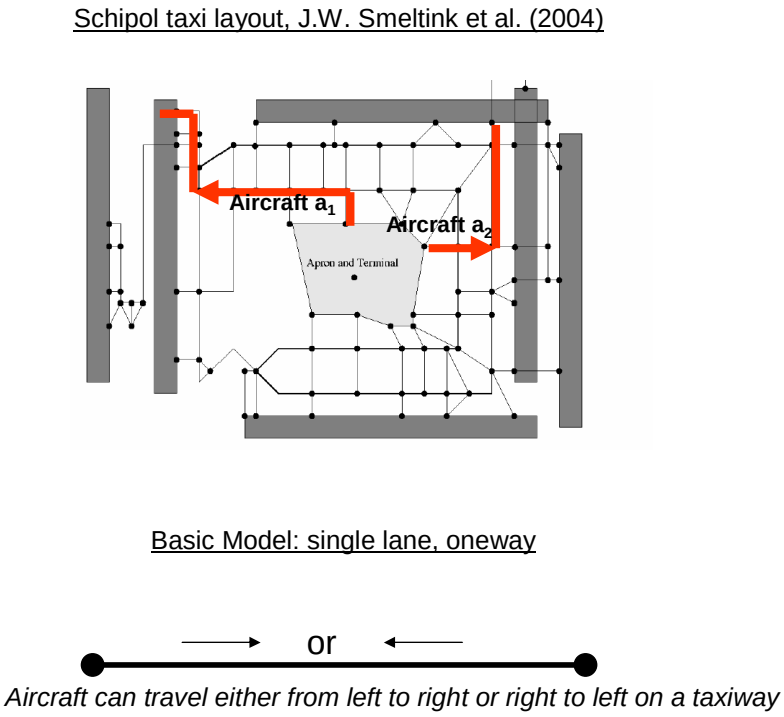

Fig. 2 In the ATP, each taxi way is a single lane, one way in which overtaking of aircraft are not allowed. The Schipol airport taxi layout is taken from Smeltink et. al. [4].

problem or its variants has been previously presented in Visser et al. [2], Smeltink et al. [4] and Balakrishnan et al. [3]. Visser et al. [2] formulate an integer program for a generalization of the taxi-scheduling problem. They address the minimum separation constraint between any two aircraft by dividing each taxi way into smaller links and allowing each link to hold at most one aircraft. For a large airport like the DFW airport, dividing each taxi link would result in a large set of variables. Also, the length of the smaller links for a given taxiway is a constant and does not depend on aircraft type. This would result in a conservative estimate on the capacity of a taxi link and does not include all the traffic conditions. For example, the length of a Boeing 737 is approximately half the length of a Boeing 777. So, a taxiway can accommodate twice the number of Boeing 737 aircraft as compared to Boeing 777.

Balakrishnan et al. [3] formulate an ILP for the ATP for the DFW airport by extending the formulation of the TFMP in Bertsimas et al. [1]. However, they do not formulate the overtaking constraints in the ILP nor discretize the taxiways into smaller cells. After solving the ILP, they propose to take care of the overtaking constraints by metering the intersection leading into each taxiway such that the aircraft when entering a taxiway are separated by the required distance. Also, the ILP they formulate has a minimum speed constraint for each aircraft on some taxiways. They use this approach and test it with data obtained from surface operations at DFW airport. However, using this approach of first finding the op-

timal solution to the ILP without the overtaking constraints (note just this step could produce optimal solutions violating overtaking constraints) and then metering the aircraft at the intersections, there is no guarantee that the aircraft sequences will be preserved on every taxiway for any input data (*i.e.*, airport layout, aircraft mix, departure time, runway crossings, etc.). For modeling purposes, the minimum speed requirement for the aircraft on some taxiways is actually not required because if the objective of the ATP is to reduce total taxi delays, then no aircraft would stop without any reason (*i.e.*, waiting for other aircraft to depart) in the optimal solution.

Smeltink et al. develop a MILP for the ATP in [4]. They retain the taxi layout of a given airport without dividing each taxiway into smaller cells. They introduce three sets of variables to formulate ATP. If aircraft i reaches an intersection u, the first set of variables track which aircraft immediately follows aircraft i to the intersection u. If aircraft i reaches an intersection u, the second set of variables track all the aircraft that follows aircraft i to the intersection u. For a given aircraft, the third set of variables denote the time instants the aircraft must reach the intersections along its route. They solve the MILP using the data collected at Amsterdam's Schipol airport and report a reduction from 20% to 2% of taxi delays. They also address issues related with uncertainty by presenting three variants of a rolling horizon algorithm.

The formulation presented in this paper is based on the formulation of Smeltink et al. [4]. However, there are two differences between the formulation presented in this paper with that of Smeltink et al. The first difference is that all the necessary minimum separation constraints are not included in [4]. Specifically, if two aircraft are traveling along the same taxiway, the Smeltink et al. approach does not include constraints for the case when the lead aircraft travels slower than the aircraft trailing behind (refer to figure 5). In this paper, it is explained how all the minimum separation constraints can be formulated using distance-time diagrams (refer to subsection 3.2). Also, the formulation in this paper is simpler with fewer variables. In particular, the set of variables that keeps track of the aircraft that immediately leads or trails an aircraft at a given taxi intersection is not used in this paper. While such variables were used in addition to other variables in [4], the taxi scheduling problem can be formulated without using them. This results in a significant reduction in the size of the problem formulation both in the number of variables and in the number of constraints.

3 Optimization model

Notations: Given a taxi layout of an airport, each taxi intersection on the layout is represented by a vertex. Each taxiway joining any two intersections is represented by a directed edge. Let $G = (V,E)$ be a directed graph with V denoting the set of all vertices and E denoting the set of edges joining the vertices. Each edge $(v_1, v_2) \in E$ joining vertices v_1 and v_2 is directed and indicates the direction that an aircraft can travel along that edge. To model the scenario when a taxiway is bi-directional (i.e., aircraft can travel in both directions along the taxiway), two directed edges are used. Let $\mathcal{D}$ and $\mathcal{A}$ be the set of all the departing and arriving aircraft respectively. Let $\mathcal{D} \bigcup \mathcal{A} = \mathcal{P}$. In this paper, only one departure runway is assumed and all arriving aircraft must cross this departure runway to reach their destinations (i.e., spots)[2]. A taxi route for aircraft i is specified by a sequence of vertices, $S_i = \{f_1^i, f_2^i, \cdots, f_{n_i}^i\}$ where f_k^i is the k^{th} vertex visited by aircraft i. Refer to Fig. 3 that shows vertex $f_{n_i}^i$ for a departing aircraft i. If vertex v is the next vertex visited by aircraft i after reaching u, then let the taxi link joining vertices u and v be denoted as (u,v). Let E_i denote the set of all taxi links present in the sequence S_i. An arriving aircraft j crosses the departure runway by traveling along edge (b_j, a_j) as shown in Fig. 3.

A departing aircraft i must reach its last vertex $f_{n_i}^i$ before its scheduled departure time (SDT_i) for take off. Let PBT_i be the push back

[2]This assumption can be relaxed and extended to other scenarios.

time for a departing aircraft i. Essentially, aircraft i is available to reach vertex f_1^i only after its push back time (PBT_i). Similarly, let SST_i and SAT_i be the scheduled destination time and scheduled arrival time for an arriving aircraft i.

Decision variables: t_{iu} denotes the time aircraft i reaches vertex u. z_{iju} is a binary integer variable. z_{iju} is equal to 1 if aircraft i visits vertex u before aircraft j and is equal to 0 otherwise. r_{ij} is equal to 1 if aircraft i uses the departure runway before aircraft j and is equal to 0 otherwise.

3.1 Objective function

The following is the objective for the taxi-scheduling problem:
Minimizing total taxi time:

$$\min \sum_{i \in P} (t_{if_{n_i}^i} - t_{if_1^i}) \tag{1}$$

3.2 Constraints

Sequencing constraints:

For all $u \in S_i \bigcap S_j$, for all $i, j \in \mathcal{P}, i \neq j$,

$$\begin{aligned} z_{iju} + z_{jiu} &= 1, \\ z_{iju} &\in \{0,1\}. \end{aligned} \tag{2}$$

Equation (2) states the requirement that if two aircraft i and j share a common vertex u along their respective paths, then either aircraft i must reach vertex u before aircraft j or vice versa.

Safety constraints:

`Overtaking constraints:`

For all $(u,v) \in E_i \bigcap E_j$, for all $i, j \in \mathcal{P}, i \neq j$,

$$z_{iju} = z_{ijv}. \tag{3}$$

If aircraft i and aircraft j share an edge (u,v) along their respective paths, equation (3) states that the sequence of the aircraft cannot be altered while traveling along the edge. This essentially implies that an aircraft cannot overtake any other aircraft while traveling on a taxi link.

`Crossing constraints:`

For all $(u,v) \in E_i, (v,u) \in E_j$, for all $i, j \in \mathcal{P}, i \neq j$,

$$z_{iju} = z_{ijv}. \tag{4}$$

If aircraft i visits vertex v after u and aircraft j visits vertex u after v along their respective paths, equation (4) states that they cannot cross each other along the taxi links connecting vertex u and vertex v.

`Minimum separation constraints at intersections:`

Let two aircraft i and j be traveling along edges (u,v) and (u,w) respectively. Vertex u is the common vertex for both the aircraft. Note that vertex v can be different from vertex w. Assume that aircraft i leads aircraft j at vertex u. Let sep_{uv}^{ij} be the minimum separation distance that the two aircraft must maintain when the lead aircraft is traveling edge (u,v). Consider the movement of the aircraft as plotted in the distance-time diagram (Fig. 4). In the figure, l_{uv} denotes the length of the edge (u,v). For the discussion given below, assume that $l_{uv} \geq sep_{uv}^{ij}$. Let l_1, as marked in the figure, denote the distance between the positions of aircraft i and aircraft j at the time instant when aircraft j has reached vertex u. It is clear the distance l_1 must be at least equal to sep_{uv}^{ij} for the minimum separation criterion to be met. This minimum separation constraint can be mathematically expressed as follows:

For all $u \in S_i \bigcap S_j, (u,v) \in E_i$ and for all $i, j \in \mathcal{P}, i \neq j$,

$$z_{iju}(l_1 - sep_{uv}^{ij}) \geq 0. \tag{5}$$

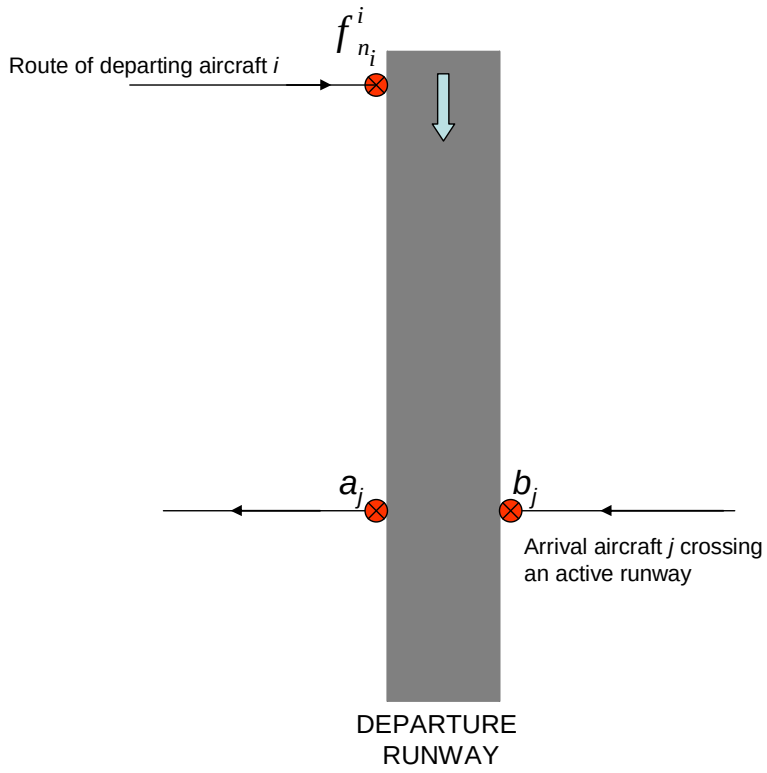

Fig. 3 An arrival aircraft j crosses the departure runway along edge (b_j, a_j).

In the above scenario, aircraft i was traveling along edge (u,v) and had already left vertex u. Consider the other scenario where aircraft i has just reached vertex v and aircraft j is traveling along the edge (w,v). In this scenario, the common vertex is v. As shown in Fig. 5, let l_2 denote the distance between the positions of aircraft i and aircraft j at the time instant when aircraft i has reached vertex v. Also, let sep^{ij}_{wv} be the minimum separation distance that the two aircraft must maintain when the trailing aircraft is traveling edge (w,v). It is clear from Fig. 5 that the distance l_2 must be at least equal to sep^{ij}_{wv}. This minimum separation constraint can be mathematically expressed as follows:

For all $v \in S_i \bigcap S_j, (w,v) \in E_j$ and for all $i, j \in \mathcal{P}, i \neq j$,

$$z_{ijv}(l_2 - sep^{ij}_{wv}) \geq 0. \tag{6}$$

Constraints stated in equation (6) are not formulated in Smeltink et al. [4].

From figure (4), one can deduce that

$$l_1 = (\frac{t_{ju} - t_{iu}}{t_{iv} - t_{iu}}) l_{uv},$$

and

$$l_2 = (\frac{t_{jv} - t_{iv}}{t_{jv} - t_{jw}}) l_{wv}.$$

Substituting for l_1 and l_2 in equations (5, 6) and linearizing them using a large positive constant M, one can obtain the following linearized equations:

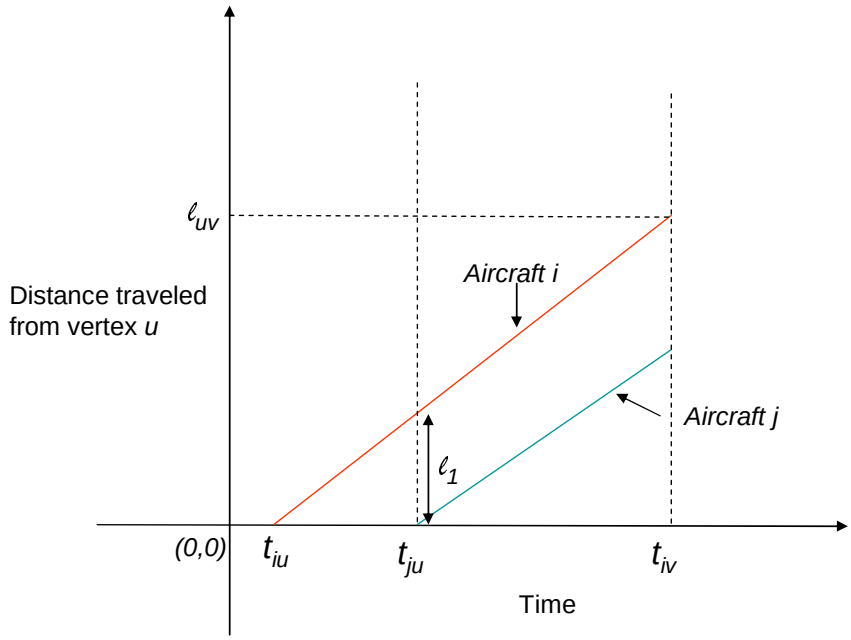

Fig. 4 Distance-time plot for two aircraft i and j. Aircraft j has just reached vertex u and aircraft i is traveling on edge (u,v).

For all $u \in S_i \bigcap S_j, (u,v) \in E_i$ and for all $i, j \in \mathcal{P}, i \neq j$,

$$t_{ju} - t_{iu} - (t_{iv} - t_{iu}) \frac{sep_{uv}^{ij}}{l_{uv}} \geq -(1 - z_{iju})M. \quad (7)$$

For all $v \in S_i \bigcap S_j, (w,v) \in E_j$ and for all $i, j \in \mathcal{P}, i \neq j$,

$$t_{jv} - t_{iv} - (t_{jv} - t_{jw}) \frac{sep_{wv}^{ij}}{l_{wv}} \geq -(1 - z_{ijv})M. \quad (8)$$

Runway occupancy constraints:

Consider any two aircraft i and j. Then either aircraft i must use the departure runway before aircraft j or aircraft j must use the runway before aircraft i. That is,

For all $i, j \in \mathcal{P}, i \neq j$,

$$r_{ij} + r_{ji} = 1. \quad (9)$$

The departure times of any two departing aircraft i and j must be at least separated by a given constant based on the type of the aircraft. This separation is required for safety reasons and is necessary to avoid one aircraft being in the wake vortex of an other aircraft. If aircraft j follows aircraft i, let the minimum separation time be denoted by T_{sep}^{ij}. This requirement is stated as follows:

For all $i, j \in \mathcal{D}, i \neq j$,

$$t_{jf_{n_j}^j} - t_{if_{n_i}^i} - T_{sep}^{ij} \geq -(1 - r_{ij})M. \quad (10)$$

If a arriving aircraft j uses the runway immediately after departing aircraft i, then the time at which aircraft j reaches vertex b_j (i.e., t_{jb_j}, refer to figure 3) must at least lag the departure time of aircraft i by a given constant (T_{dep}). This requirement is stated as follows:

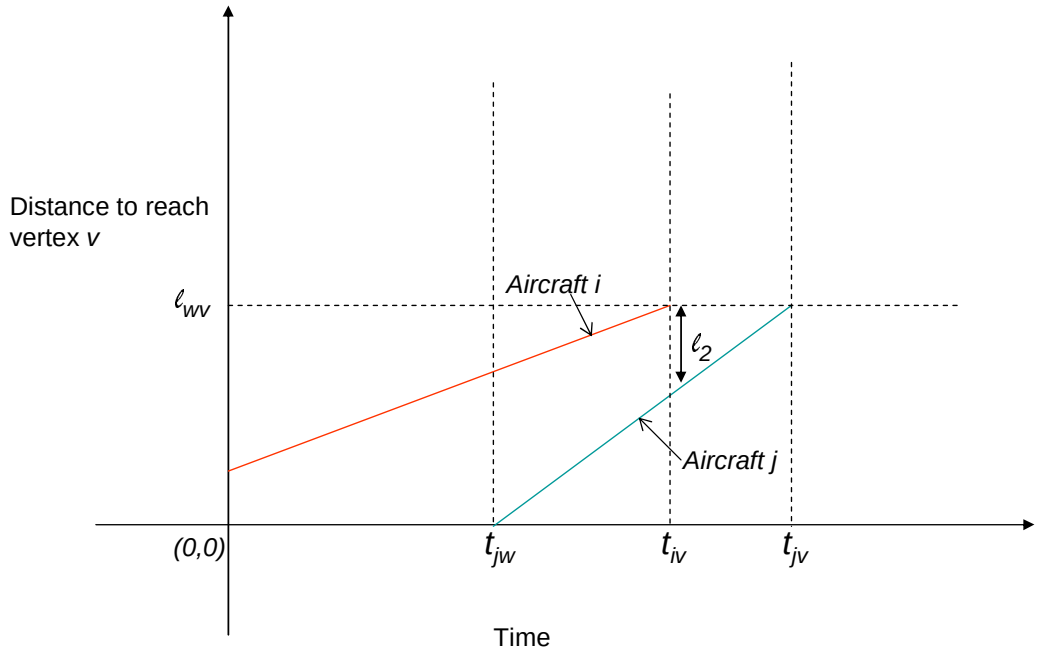

Fig. 5 Distance-time plot for two aircraft i and j. Aircraft i has just reached vertex v and aircraft j is traveling on edge (w,v).

For all $i \in \mathcal{D}, j \in \mathcal{A}$,

$$t_{jb_j} - t_{if^i_{n_i}} - T_{dep} \geq -(1-r_{ij})M. \quad (11)$$

Similarly, if an departing aircraft i uses the runway immediately after an arriving aircraft j, then the departure time of aircraft i ($t_{if^i_{n_i}}$) must at least be equal to the time at which aircraft j reaches vertex a_j (t_{ja_j}). This requirement is stated as follows:

For all $i \in \mathcal{D}, j \in \mathcal{A}$,

$$t_{if^i_{n_i}} - t_{ja_j} \geq -(1-r_{ji})M. \quad (12)$$

Origin and destination timing constraints:

For all $i \in \mathcal{D}$,

$$t_{if^i_{n_i}} \leq SDT_i,$$

$$t_{if^i_1} \geq PBT_i. \quad (13)$$

For all $i \in \mathcal{A}$,

$$t_{if^i_{n_i}} \leq SST_i,$$
$$t_{if^i_1} = SAT_i. \quad (14)$$

The above equations state that a departing aircraft i can reach the first vertex along its path after its given push back time (PBT_i) and must reach the last vertex along its path by its scheduled departure time (SDT_i). Similarly, an arriving aircraft i reaches the first vertex at the given arrival time (SAT_i) and must reach its last vertex by its scheduled spot time (SST_i).

Speed constraints:

For all $(u,v) \in E_i$, for all $i \in \mathcal{P}$,

$$\frac{l_{uv}}{v_{uv}^{max}} \leq t_{iv} - t_{iu}. \quad (15)$$

In the above equation l_{uv} denotes the length of edge (u,v). v_{uv}^{max} is the maximum speed that any aircraft can travel along edge (u,v).

Remarks:

1. The above optimization model includes two difficult problems, namely the sequencing problem and the time scheduling problem. If the sequence of all the aircraft at each taxi intersection is given, then the above formulation reduces to a linear program in just the continuous variables.

2. Note that there is no need for separate modeling of queues. It is automatically taken care of by the minimum separation constraints.

3. The objective in the problem minimizes the total taxi time of all the aircraft. Other objectives such as minimizing the maximum destination time of any aircraft or minimizing the total deviations from a given destination time for each aircraft can also be formulated.

4. In the simpler case when only departure aircraft are considered, this problem is a direct generalization of the well known single machine scheduling problem with job dependent separation constraints [5].

4 Simulation results

As a preliminary case study, a set of 25 aircraft (12 large, 13 heavy) departing from the east side of DFW airport has been considered. A set of 25 aircraft was chosen because normally during rush hours, around 25 aircraft depart from the east side of the DFW airport in 30 minutes. Fig. 6 shows the network layout of the east side of the DFW airport. Each aircraft is available at a specified spot after its push back time. A route is assigned to each aircraft from its spot to the departure runway.

Fig. 6 shows a sample route for a departure aircraft. The maximum speed of an aircraft on any taxiway is assumed to be 8 m/sec. The minimum separation distance between any two aircraft must be at least 200 meters at any time instant and is a constant for all the taxiways. The departure timings of all the aircraft must be separated based on their type due to the wake vortex separation constraints.

The separation matrix given in Fig. 7 was used for the simulations. For example, if a heavy aircraft follows a large aircraft on a departure runway, then their departure times must be separated by at least 75 seconds. The push back times of all the aircraft are randomly chosen using a uniform distribution in a time window $[0,T]$.

To simulate a case with high congestion, T can be chosen to be a small number. Similarly, choosing a large value for T would result in a case with low congestion. For the simulation study, six values for T were considered: $T = 0,5,10,15,20,25$. For each $T > 0$, 100 scenarios were generated by choosing the push back times of each aircraft uniformly in the interval $[0,T]$. For $T = 0$, there is only one scenario as all the aircraft are available at their respective spots at time equal to 0.

Optimizing the push back time of the departure aircraft could result in solutions that would hold the departure aircraft as long as possible at the spots. Therefore, each aircraft in the optimization model is constrained to only wait for a maximum of 10 minutes in all the simulations.

To assess the benefits of optimization, a FCFS algorithm has also been implemented as a base case. At any time instant, the FCFS algorithm always gives priority to the aircraft that reaches a taxi intersection or the departure runway at the earliest time. The FCFS algorithm also enforces the maximum speed constraint and maintains the minimum separation between any two aircraft at any time instant. The FCFS algorithm generates both the departure sequence at the runway and the times at which all the

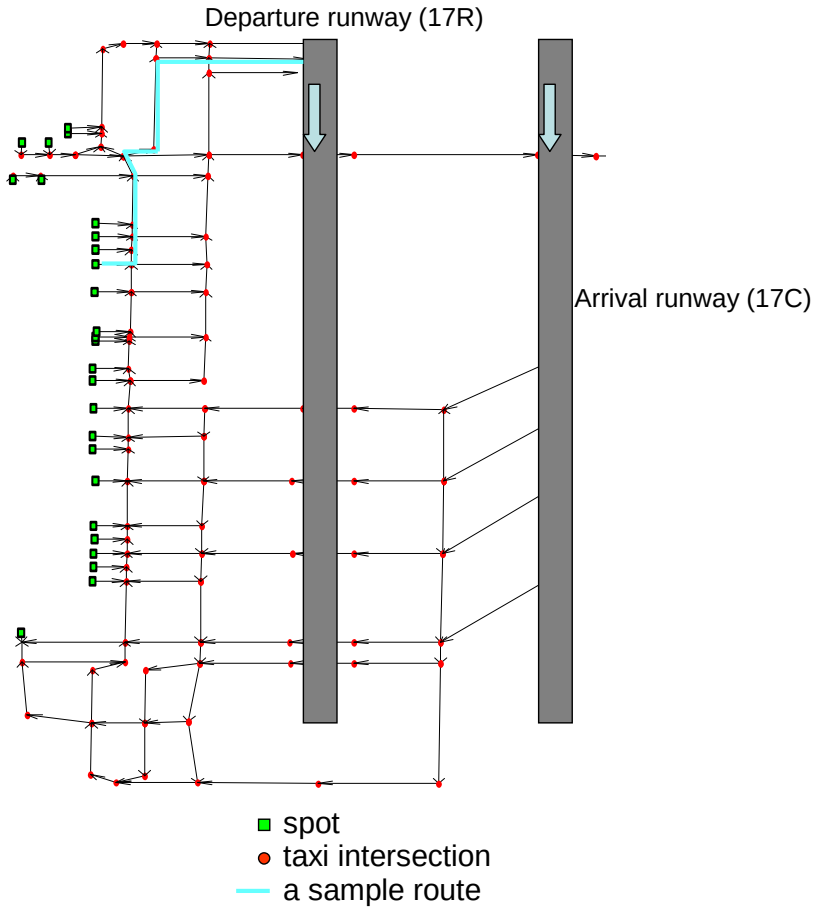

Fig. 6 Network layout of the east side of the DFW airport.

Leading aircraft

Trailing aircraft	Large	Heavy	B757
Large	55	110	90
Heavy	75	100	75
B757	55	110	60

Fig. 7 Separation matrix for departure aircraft in seconds.

aircraft reach their respective taxi intersections. The average taxi time per aircraft generated by the FCFS algorithm for different values of T is shown in Fig. 8. For example, if T is equal to 15 minutes, the average taxi time per aircraft is 14.14 minutes for the FCFS algorithm.

Once the departure sequence is determined using the FCFS algorithm, it is then used as an input for the optimization model of the taxi scheduling problem. The basic idea here is to compare the taxi times obtained by the FCFS algorithm with those from the optimization model given the departure sequence of the aircraft on the runway. To simplify the MILP formulation, the departure sequence at the runway is used for each of the taxi intersections for all the aircraft. As a result, the optimization model reduces to a linear program where there are only continuous variables (i.e. times at which the aircraft must reach all their respective intersections).

This linear program was programmed in MATLAB [6] and solved using the YALMIP tool box [7]. YALMIP is an open source software available for MATLAB and can be used to solve MILPs. Fig. 8 shows the average taxi time per aircraft as obtained by solving the optimization model. Fig. 9 shows the average savings in taxi time using the optimization model. For the I^{th} scenario, let $Cost^I_{fcfs}$ ($Cost^I_{opt}$) be the total taxi time obtained by using the FCFS algorithm (optimization model). The percentage savings using the optimization model with respect to the FCFS algorithm for a given scenario (I) is defined as $\frac{100(Cost^I_{fcfs}-Cost^I_{opt})}{Cost^I_{opt}}$. Figure (10) shows the average savings in percentage. Though the average savings for different values of T is around 6 minutes, the savings in percentage decreases as T tends to zero. This is due to the fact that the difference between $Cost^I_{fcfs}$ and $Cost^I_{opt}$ decreases as the maximum pushback time, T, decreases. As congestion increases, this says that the taxi times using the first come first serve algorithm is similar to an optimal solution.

5 Conclusions

An optimization model for an aircraft taxi scheduling problem has been presented in this paper. The proposed model improves on the formulation presented by Smeltink et al. [4]. Specifically, the model presented includes all of the safety avoidance constraints and is a simpler formulation with a significant reduction in the number of variables and constraints. To assess the benefits of optimization, a FCFS algorithm has also been implemented. For a preliminary study involving 25 departing aircraft, on average, approximately six minutes of taxi time were saved per aircraft using the optimization model in comparison with the FCFS algorithm. Currently, the authors are investigating the scenarios where both departure and arrival aircraft are present. Fast heuristics are also being developed for the taxi scheduling problem with an aim of producing quality solutions that can be computed efficiently.

6 Acknowledgement

The authors thank Kenneth Kuhn, Jeffery Schroeder, Banavar Sridhar, Katharine Lee, Mike Madson, Avijit Mukherjee and Thomas Davis for their useful suggestions on the earlier drafts of this paper.

References

[1] Bertsimas D.J. and Stock Patterson, S., “The air traffic flow management problem with en-route capacities”, Operations Research, Vol. 46, pp. 406-422, 1998.

[2] Visser, H. G. and Roling, P. C., “Optimal Airport Surface Traffic Planning Using Mixed Integer Linear Programming", AIAA Aviation Technology, Integration and Operations (ATIO) Conference, Denver, CO, 2003.

[3] Balakrishnan, H., and Jung, Y., “A Framework for Coordinated Surface Operations Planning at Dallas-Fort Worth International Airport”, AIAA Guidance, Navigation, and Control Conference, Hilton Head, SC, August 20-23, 2007.

[4] Smeltink, J.W., Soomer, M.J, De Waal, P.R. and

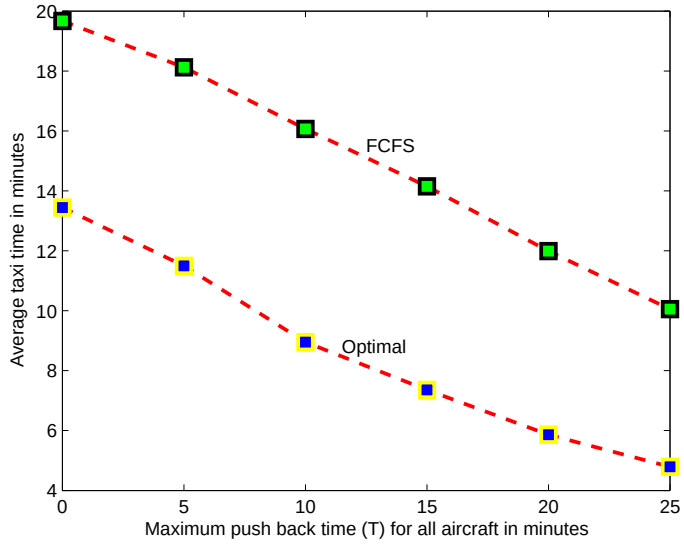

Fig. 8 Average taxi time per aircraft obtained using the optimal solution and the FCFS algorithm.

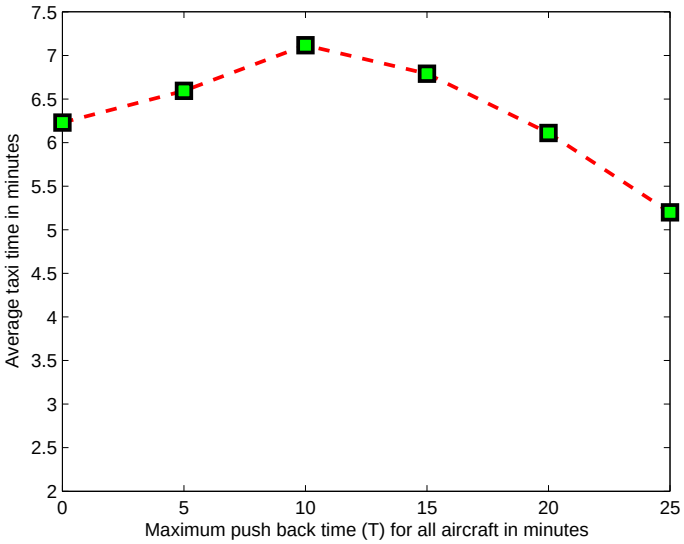

Fig. 9 Savings in the average taxi time per aircraft using the optimal solution.

Van Der Mei, R.D., "An Optimisation Model for Airport Taxi Scheduling", Thirtieth Conference on the Mathematics of Operations Research, Lunteren, The Netherlands, January 2005.

[5] Brucker, P., and Knust, S., Complexity Results for Single-Machine Problems with Positive Finish-Start Time-Lags, Computing 63(4): 299-316 (1999).

[6] Matlab, The Language of Technical Computing, The Math Works, Inc.

[7] YALMIP : A Toolbox for Modeling and Optimization in MATLAB. J. Löfberg. In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.

Copyright Statement

The authors confirm that they, and/or their company or institution, hold copyright of all original material included in their paper. They also confirm they have obtained permission, from the

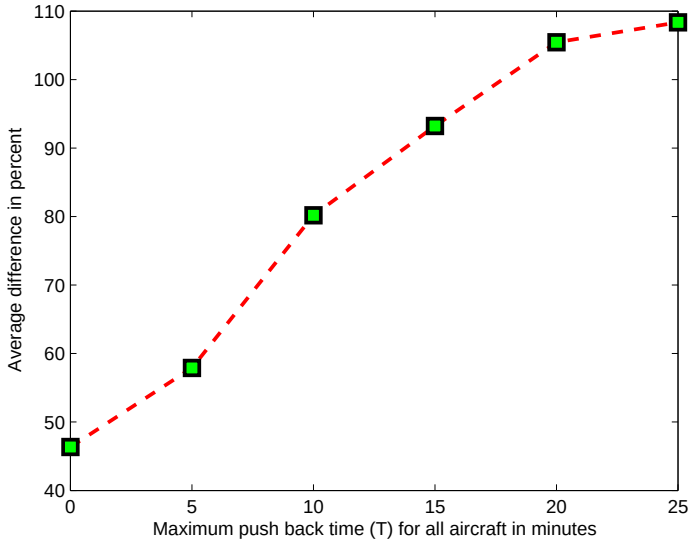

Fig. 10 Average difference in percent using the optimal solution with respect to the FCFS algorithm.

copyright holder of any third party material included in their paper, to publish it as part of their paper. The authors grant full permission for the publication and distribution of their paper as part of the ICAS2008 proceedings or as individual off-prints from the proceedings.