Runway Operations Planning:

A Two-Stage Heuristic Algorithm

Ioannis Anagnostakis[*], yianag@mit.edu
Prof. John-Paul Clarke[†], johnpaul@mit.edu
Dept. of Aeronautics & Astronautics, Massachusetts Institute of Technology
77 Massachusetts Avenue, Room 35-217, Cambridge, MA 02139, USA
Tel.: 617-253-3507

Abstract

The airport runway is a scarce resource that must be shared by different runway operations (arrivals, departures and runway crossings). Given the possible sequences of runway events, careful Runway Operations Planning (ROP) is required if runway utilization is to be maximized. From the perspective of departures, ROP solutions are aircraft departure schedules developed by optimally allocating runway time for departures given the time required for arrivals and crossings. In addition to the obvious objective of maximizing throughput, other objectives, such as guaranteeing fairness and minimizing environmental impact, can also be incorporated into the ROP solution subject to constraints introduced by Air Traffic Control (ATC) procedures. This paper introduces a "two stage" heuristic algorithm for solving the Runway Operations Planning (ROP) problem. In the first stage, sequences of departure class slots and runway crossings slots are generated and ranked based on departure runway throughput under stochastic conditions. In the second stage, the departure class slots are populated with specific flights from the pool of available aircraft, by solving an integer program with a Branch & Bound algorithm implementation. Preliminary results from this implementation of the two-stage algorithm on real-world traffic data are presented.

1. Introduction

Significant delays have been observed in the departure flow at many major European and US airports. Most of these delays occur at the takeoff queue next to the runway, where aircraft line up with their engines running waiting for a slot on the active runway. Similar delays occur during other phasesof the taxi out process, i.e. when poor planning results in excessively long waits (with the engines running) at intersections and/or ramps. These delays result in economic (higher fuel costs) and environmental (higher emissions) inefficiencies.

In order to mitigate the adverse economic and environmental effects of ground congestion and delays it is critical that:

Runway efficiency is improved,
Runway queue delays are minimized,
Taxi out times are minimized and
Engine run times are minimized.

Airport departure management requires that air traffic controllers perform several control tasks, e.g. pushback, "engine start", taxiway entry, runway assignment and takeoff clearances. In many instances, these tasks must be performed under conditions of high workload and time criticality. In addition, observations of operations at airports such as Boston Logan [14], [15], [16], Washington Dulles [3] and Newark [8], indicate that the dynamics of airport ground flows heavily depend on Air Traffic Control (ATC) constraints and how these affect each airport site [13], [14]. Thus, given the complexity of the departure process and the site specific nature of departure operations, it is difficult for controllers to fully explore all the possible solutions within the relatively short time period in which decisions must be made.

NASA-sponsored research on causes of departure delay [2], [4], [14], [19] all suggest that an

[*] Graduate Research assistant, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Student Member AIAA
[†] Associate Professor of Aeronautics & Astronautics, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Senior Member AIAA

automation aid to help optimize and control the departure flow would benefit both controllers and aircraft operators. It is no surprise then that recent research in the field of airport surface operations has focused on decision aiding technologies such as the Surface Movement Advisor (SMA) [11], [12], [18] and the Departure Enhanced Planning And Runway/Taxiway-Assignment System (DEPARTS) [9], [10]. In fact, the primary objective of the Surface Management System (SMS) research prototype being developed by NASA is to contribute to the understanding and solution of various problems existing on the surface of airports within the National Airspace System [5], [6], [21].

It is very likely then that an automated decision support system, which includes planning and control algorithms, will be a significant component of the solution to any surface optimization problem. Such systems will both explore a very large number of possible future departure schedules to determine the optimal schedule and reduce uncertainties by exercising tighter sequencing and scheduling control on each portion of the departure process.

This paper introduces a "two stage" optimization algorithm for solving the Runway Operations Planning (ROP) problem i.e. to determine the optimal departure schedule. The paper is organized as follows: The structure of the proposed "two-stage" algorithm is described in Section 2. The methods used to model and implement the algorithm, including the Matlab-Simulink model created to test the behavior of the algorithm, are described in Section 3. Preliminary results for a typical airport geometry are presented in Section 4. A short summary together with the goals for future work in this area in presented in Section 5.

2. *Runway Operations Planning*

2.1. Definition

The goal of runway operations planning is to generate a schedule of operations (arrivals, departures, and crossings) that are as close to optimality as possible while taking into account uncertainties in pushback and taxi operations. Successful implementation of these optimal schedules will minimize departure inefficiencies related to such factors as wake vortices, downstream constraints (splitting departure routes, jet-prop mix, arrival & departure mix), workload limitations, and intersecting runways.

2.2. A "two-stage" algorithmic approach

The methodology outlined in [1], focused on optimally solving the ROP problem in a *"one-stage" optimization routine* that considers all the characteristics of each aircraft (e.g. weight class, destination, ATC constraints) at the same time. The methodology presented here differs from that approach in a very significant way. Specifically, the optimal runway operations schedule is determined by *parsing the problem into two simpler stages*, as depicted in the flow chart in Figure 1 and incorporating the various objectives and constraints at the most appropriate stages.

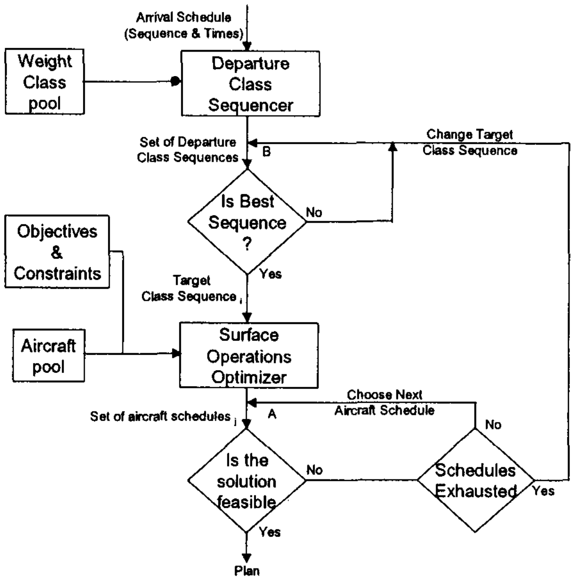

Figure 1: Optimization in stages

The two-stage algorithm for runway operations planning given a fixed arrival schedule may be summarized as follows:

- **Stage 1 - Departure Class Sequencer:** The arrival sequence and touchdown times are assumed to be known some time in advance. Some or all of these arrivals, after landing and decelerating on their runway, generate runway-crossing requests on other runways. After the earliest possible times for crossing requests are determined, time slots on the departure runway are allocated to departing aircraft and crossing aircraft (and arriving aircraft if that runway is used for arrivals and departures). This allocation is performed with respect to the *single* objective of maximizing throughput. The result is a matrix CS of Departure Class Schedules each of which is a sequence of specific weight classes but NOT specific aircraft or flights:

  CS = [Class Sequence 1,

  •

  Class Sequence i,

  •

  Class Sequence m]

The sequences are ordered with respect to their corresponding throughput. Since it is possible that more than one class schedules can have the same runway throughput, several class sequences might have optimal (maximum) throughput.

- **Stage 2 - Surface Operations Optimizer**: The best class sequence (in terms of throughput) from the first stage is defined as the "Target Class Sequence" (TCS). Each weight class slot in the TCS is then assigned an aircraft of the specified weight class (an "aircraft – to – slot" assignment) while satisfying all or as many as possible of the remaining system objectives (e.g. delays and environmental impact, fairness). The output is a matrix AS of aircraft schedules:

  AS = [Aircraft Schedule 1,
  •
  Aircraft Schedule j
  •
  Aircraft Schedule n]

  If the best aircraft schedule is not feasible because it violates "hard" (inviolable) system constraints[1] such as ATC restrictions, the next possible aircraft schedule is chosen (feedback A in Figure 1). If all possible aircraft schedules are exhausted i.e. none of them is feasible, the Target Class Sequence is changed to the next available class sequence from the output matrix generated by the Departure Class Sequencer (feedback B in Figure 1).

At the most fundamental level, both stages perform the two functions required to determine the optimal sequence. The class of each departure slot is defined in the first stage, and specific aircraft are assigned to each of the defined class slots in the second stage. While the second stage may be performed immediately after the first, the two stages may also be performed separately depending on the needs of the particular real-world situation.

For example, assume that both stages of the algorithm have been performed and a schedule with specific aircraft for each class slot has been generated. If one or more of these aircraft have difficulty meeting that schedule, the class slot sequence generated in the first stage can be left untouched if it is too costly or impractical to change it, while the second stage (specific aircraft assignment) can be performed independently to assign new flights to substitute for those flights that are unable to meet their class slots.

In this way, the time scale and level of control in each stage is better matched to the dynamics of the situation. The first stage of the algorithm provides control over the throughput aspects of the optimization problem, while tighter departure control might be achieved in the second stage of the algorithm though the scheduling of release times and takeoff times for specific aircraft.

For the remainder of the paper, the terms "(departure) class schedule" and "(departure) class sequence" will be used interchangeably. Similarly, the terms "(departure) aircraft schedule" and "(departure) aircraft sequence" will also be used interchangeably.

3. *Algorithm Implementation*

The "two-stage" algorithm described in Section 2 was implemented using Matlab and Simulink. The rationale for this choice is that Matlab offers a robust programming environment, which also supports data structures, and the combination of Matlab and Simulink offers rapid prototyping capabilities and modularity. Thus, it is easy to make changes to the airport surface and runway geometry being modeled. Figure 2 presents an overview of the Simulink test model in its current state.

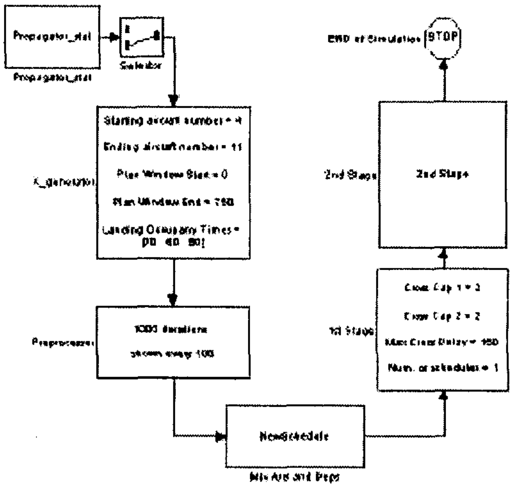

Figure 2: Simulink Model Overview

The model input is in the form of schedule files that contain arrival and departure data structures. These data structures list the details of the flights that are scheduled to be operated in and out of the airport. The departure part of these files has the format:

```
DEPAC(12).Simulation.Source='Terminal A';
DEPAC(12).Parameter.Type='J';
DEPAC(12).Parameter.Class='L';
DEPAC(12).Parameter.Model='B737';
DEPAC(12).FlightPlan.Destination='ATL';
```

[1] For a definition of "hard" constraints, see [1]

```
    DEPAC(12).Simulation.EntryTime=[105];
    DEPAC(12).Identification.CallSign='DL182';
    DEPAC(13).Simulation.Source='Terminal A';
    DEPAC(13).Parameter.Type='J';
```

and the arrival part has the format:

```
    ARRAC(1).Simulation.Source='Fix A';
    ARRAC(1).Parameter.Type='J';
    ARRAC(1).Parameter.Class='H';
    ARRAC(1).Simulation.LandingTime=[75];
    ARRAC(2).Simulation.Source='Fix A';
    ARRAC(2).Parameter.Type='P';
```

Each field contains the characteristics of a departing or arriving flight that are essential to the model. The schedule files allow the user to add new characteristics if necessary, depending on how the design and the demands of the model evolve. In real-world applications, these schedule files might possibly be automatically populated through a schedule feed from the ATC Host computer.

The planning window that the algorithm works with can either be time-based or aircraft-based (i.e. contain a certain number of aircraft from the schedule for which runway times will be planned). Currently, depending on the weight class homogeneity of the group of aircraft to be planned, the Matlab model is only able to handle aircraft groups up to a certain size, mostly due to computer memory limitations. Therefore, depending on the density of the schedule, a time-based window may result in a total number of aircraft that is too great for the model to handle. For this reason, aircraft-based planning windows are used.

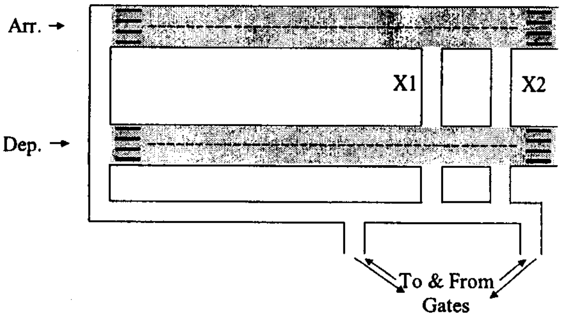

Figure 3: Example airport system

Figure 3 depicts a hypothetical airport system with two parallel runways, one dedicated to arrivals and the other to departures. Aircraft landing on the arrival runway eventually gather at one of the two cross-points X1 or X2 and wait for clearance to cross the departure runway on the way to their gates. Departures also interact with arrivals on the taxiways, ramps and gate areas, as they taxi to the departure runway for takeoff. Since this geometry is frequently encountered at US airports, it will be used as the reference geometry throughout this paper.

3.1. Stage One

The sole objective of the first stage is to determine the best departure class sequence (from a throughput perspective) to be used in the second stage. This is achieved by calculating the throughput for each class sequence in a family of "*enriched*" class sequences. The term "enriched" is used to denote the fact that the departure class sequences also include "place holders" for aircraft crossing the departure runway, as well as for arriving aircraft if dual operations are performed on the runway. The entire process is described below.

The list of possible class schedules is first determined using a "*preprocessor.*" As is to be expected, the number of possible class schedules to consider depends on the number N_D of departures involved. Specifically, if the N_D departures consist of N_{DH} heavy aircraft, N_{DL} large and N_{DS} small aircraft, the number of possible schedules is:

$$\frac{N_D!}{N_{DH}!\cdot N_{DL}!\cdot N_{DS}!}$$

In most practical instances, this number will be very large. For example, assume that there are twenty (20) aircraft within the planning period and that the corresponding weight classes are: three heavies (H), twelve large (L) and five small (S). The effective number of possible class sequences is equal to 20! / (3!*12!*5!) = 7054320. For this reason, a random seeding procedure was developed and implemented. In this procedure, a specified number of randomly generated sequences are created and then pruned to eliminate repetitions. The number of randomly generated sequences required to adequately explore the solution space was determined through a series of sensitivity analyses to be on the order of 1,000.

The departing aircraft under consideration are then propagated from their gates to the runway to obtain an initial estimate of the runway time window that these aircraft will occupy. The probability density function for the taxi time is assumed to be a normal distribution with appropriate mean and standard deviation. These parameters (mean and standard deviation) may be derived using one of the following methods:

- *Method 1*: Analyzing approximately four (4) hours of ground traffic data obtained through observations at the Boston Logan airport control tower on February 2, 1999 [16] to determine the taxi time for each ramp and taxiway segment.
- *Method 2*: Fitting the prototype taxi model

introduced in [17] and implemented in [7] to Airline Service Quality Performance (ASQP) data for Logan airport and using the model to predict the unimpeded taxi time from the gate to the runway.

- *Method 3*: Analyzing ASQP data from nighttime operations when traffic levels are lower to estimate the unimpeded taxi time from the gate to the runway.

Each of these methods has its limitations and drawbacks. The first method requires an extensive data set that is difficult to collect. The second and third methods do not provide taxi time information for individual ramp and taxiway segments and at most airports, the lack of information about runway configuration and gate used by each aircraft limits the fidelity of the results that are obtained.

The runway time windows are then converted into feasibility constraints on the number of aircraft of different classes that will be available during the time period under consideration. These feasibility constraints are used to further reduce the list of possible class schedules. The throughput of each departure class sequence in the list of possible class sequences is then determined by applying ATC rules regarding the time between successive departures.

At this point, the departure class schedules are "enriched" i.e. runway crossings are injected into the departure schedule using two heuristics. The two heuristics may be described as follows:

- Runway crossings should be done in "groups" i.e. multiple crossings should be performed at the same time. This is desirable because the time required to complete multiple crossings is less than the total time required to cross each aircraft individually. And, is based on the assumption that there are a number of aircraft waiting to depart so that no time slot will go unused.
- Runway crossings should be scheduled after "heavy" departures if there are heavy aircraft in the pool of departing aircraft. This is desirable because the required wake vortex separation time behind a heavy departure is larger than the required wake vortex separation behind an aircraft of any other weight class. And, for the typical airport geometry, this time is equal to or grater than the time required to perform multiple crossings.

Both of these heuristics are subject to constraints on the number of aircraft that may be held at runway crossing points (due to physical limitations) and on the maximum ground delay that may be imposed on arriving aircraft.

In all the cases tested so far, the process of adding crossings to the departure schedules has also served to further reduce the number of possible class schedules. This is because some of the original class schedules (without crossings) are eliminated from further testing because they are likely to introduce excessive amounts of delay to all aircraft. This effect may be explained as follows. The model finds the weight class of the first class slot of each schedule. Then, it selects among the available aircraft of the same weight class the one that is expected to reach the runway earliest. If that aircraft is not also the earliest among all available aircraft, then the class schedule is not acceptable. Assume, for example, that the first slot of a class schedule is a "large" (L) class slot. Also, assume that among all large aircraft, aircraft X is the one that is expected to reach the runway first. If aircraft X is also the aircraft that is expected to arrive at the runway first among all available aircraft (irrespective of their weight class) then the schedule is acceptable. If that is not the case and (for example) a small aircraft is expected to arrive at the runway first, then that small aircraft would have to absorb high delays while waiting for one of the large aircraft to occupy the first slot of the class schedule. Therefore, in this case, the schedule is eliminated because it would result in high aircraft delays.

After generating departure class sequences that include runway crossings, the throughput of each class schedule is updated to reflect the fact that crossings usually increase the amount of time it takes for a set of departures to be completed. In calculating throughput values for each class schedule, the stochastic nature of ground operations leaves no choice but to calculate stochastic throughput using probabilistic distributions for the pushback and taxi processes. Using as a "base" schedule one of the departure class schedules with crossings, these distributions help determine the probability of a class slot actually being at the position it has in the "base" schedule, as opposed to occupying one position up or down in the sequence (shifts of only one position were allowed for simplicity). For each "base" schedule, its final stochastic throughput is calculated as the expected throughput over all the possible schedules that can be derived from the "base" by performing feasible class slot shifts.

For example, assume that there are fourteen (14) departures within the predetermined planning window. Also, assume that it has been estimated that these departing aircraft are expected to interact with four (4) arrivals, which will request runway time in order to cross the departure runway. Therefore, a possible departure class schedule including crossings (crossing are denoted by lowercase letters) is:

L – L – L - H - **s/s/h** – H - **l** - L – L – L – L - L –L – S – S - S

In this schedule, which is considered to be the "base" schedule, there are only three (3) possible class slot (one-position) swaps that can actually affect the throughput of this sequence, as shown in Figure 4 (X1 and X2 are abbreviations for the two crossings groups). Taking all possible combinations of occurrence of these three swaps, the set of possible class schedules that can be derived from the "base" schedule consists of eight (2^3) schedules (including the "base"), which are shown in Figure 5.

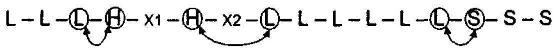

Figure 4: Possible class slot swaps that affect throughput

L – L – L – H – X1 – H – X2 – L – L – L – L – L – L – S – S – S
L – L – L – H – X1 – H – X2 – L – L – L – L – L – **S** – **L** – S – S
L – L – L – H – X1 – **L** – X2 – **H** – L – L – L – L – L – S – S – S
L – L – L – H – X1 – **L** – X2 – **H** – L – L – L – L – **S** – **L** – S – S
L – L – **H** – **L** – X1 – H – X2 – L – L – L – L – L – L – S – S – S
L – L – **H** – **L** – X1 – H – X2 – L – L – L – L – L – **S** – **L** – S – S
L – L – **H** – **L** – X1 – **L** – X2 – **H** – L – L – L – L – L – S – S – S
L – L – **H** – **L** – X1 – **L** – X2 – **H** – L – L – L – L – **S** – **L** – S – S

Figure 5: Schedules derived from the "base" schedule by performing all possible swap combinations

The throughput for each of the derived schedules is calculated based on the normal distributions for pushback and taxi time. For each schedule, the mean value for the starting time-point of the first class slot is the mean "Time at the Runway" for the earliest aircraft in the departure pool that has the same weight class as the starting class slot. In this example, this would be the time at the runway for the earliest large (L) aircraft. If the pushback process and the remainder of the taxi process are assumed to be independent, the runway time may be calculated as the sum of the mean pushback time (including pushback delays) and the mean taxi time (from gate to runway threshold) for that specific aircraft and for the specific terminal it is coming from.

Once the time of the class slot for the first departure is determined, the times of the other class slots can be determined from wake vortex separation criteria and the duration of the activity in each slot. Then, we can assign a probabilistic curve to each slot using the starting point or middle point of the slot as the "mean value" and the taxi time standard deviation as the standard deviation. The overlapping regions between curves of adjacent class slots, determine the probability of a swap between those two slots occurring. Based on those swap probabilities and the combination of swaps involved in each derived schedule, a probability of occurrence and a throughput value for that particular derived schedule can be calculated. The final stochastic throughput for the "base" schedule is calculated as the expected throughput over the throughput values of all the derived schedules, each of them considered with its individual probability of occurrence.

This process is repeated for each class schedule with crossings, and finally the list is ordered according to throughput in descending order. The first few departure class schedules with crossings are then considered to be the best in terms of maximizing throughput and therefore, they are candidates to become the "Target Class Schedule" in the second stage of the algorithm.

3.2. Stage Two

The second stage optimization is formulated as an integer program that generates a solution that represents the assignment of aircraft to class slots. For that reason, the decision variables selected for the formulation of the integer program were chosen to be X_{ij}, where $X_{ij} = 1$ if aircraft i occupies slot j, and $X_{ij} = 0$ otherwise.

Objective Function

Given that throughput maximization is addressed in the first stage of the algorithm, a delay-based objective function is used to address the remaining constraints. The time assigned to each runway event is set equal to the midpoint of the time slot to which the specified aircraft is assigned. For example, if the Target Class Schedule with crossings is:

L – L – H - **s/s/h** - L – S – S – H - **l** - L –L –S

and the absolute earliest time that a large aircraft from the departure pool can be at the runway is estimated to be 670, then the following set of times corresponds to the midpoints of the slots in the Target Class Schedule, based on wake vortex separations and landing / crossing runway occupancies (the first crossing aircraft assumed to occupy the runway for 40 sec and each aircraft behind it for an additional 10 sec):

700 – 760 – 820 – **(X)** - 930 – 990 – 1050 – 1110 – **(X)** - 1230 – 1290 – 1350

For the general case of a runway that serves all types of operations and alterations to the arrival schedule are permitted, let the original arrival (touchdown) times be TOn_i, the projected crossing request times of those arrivals be TX_i and the target departure (clearance to takeoff) times be the class slot midpoint values $TOff_j$ that are calculated. For every arrival i, $1 \leq i \leq N_A$, where N_A is the total number of arrivals considered and for every departure j, $1 \leq j \leq N_D$, where N_D is the total number of departures considered. $N_A + N_D = N$, is the total number of "mixed" operations on the runway(s) during the current scheduling window. If only departures and crossings are serviced on the runway, then $N_A = 0$.

The delay for each operation is defined as the difference between actual touchdown, crossing or takeoff time and the corresponding earliest possible values for each flight EOn_i, EX_i and $EOff_j$. The latter are calculated using the input arrival and departure schedules and the unimpeded taxi time values derived earlier in the process. Hence, the delay value for each operation represents how much later than the earliest possible time an operation will occur. The total delay for the runway (i.e. minimum arrival, departure and crossing delay) can then be formulated as:

Min aggregate delay:

$$\min\left(\sum_{i=1}^{N_D}\left|TOff_i(x_i)-EOff_i\right|^{k_D}+\sum_{j=1}^{N_A}\left|TOn_j-EOn_j\right|^{k_A}+\sum_{m=1}^{N_A}\left|TX_m-EX_m\right|^{k_X}\right)$$

where $1 \le i \le N_D$ and $1 \le j,m \le N_A$

or

Minimize ONLY departure delays:

$$\min\sum_{i=1}^{N_D}\left|TOff_i(x_i)-EOff_i\right|^{k_D}$$

where $1 \le i \le N_D$, x_i is the slot position of aircraft i and k_A, k_D and k_X are parameters used to penalize delays of specific flights, with $k_A \ge 1, k_D \ge 1$ *and* $k_X \ge 1$.

Constraints

The most fundamental operational constraint that is satisfied in the first stage of the algorithm is the *wake vortex* separation standards. Since the problem formulation also has to reflect the fact that physical constraints will constrain aircraft movement, there will be some class slots in the Target Class Schedule (TCS) that certain aircraft cannot "make" (occupy) in the problem solution. For example, if the earliest time that aircraft i is expected to be at the runway is time 900 and the time at the midpoint of the first two slots in the TCS is earlier than time 900, aircraft i cannot be allowed to occupy slots 1 and 2 in the final solution. This type of constraint can be easily formulated as $X_{ij} = 0, \text{for } j = 1,2$.

The class *slot sequence* of the Target Class Schedule also has to be enforced. So, for example, in a TCS with10 slots, if slots 2,3, 4, 5 and 6 are the "large" slots in the sequence and aircraft i is the only non-large aircraft in the aircraft pool, the following has to be true in the final solution: $X_{1j} = 0, \forall\, j \in [2, 3, 4, 5, 6]$. This can be guaranteed by setting the constraint $\sum_j X_{1j} = 1,\ \forall\ slot\ j \in [1,7,8,9,10]$.

Additionally, each aircraft must occupy only one slot $\sum_{j=1}^{N_S} X_{ij} = 1,\ \forall\ aircraft\ i$, where N_S is the total number of slots in the class slot sequence, and each slot must be occupied by only one aircraft $\sum_{i=1}^{N_D} X_{ij} = 1,\ \forall\ slot\ j$.

In many cases, maintaining departure fairness among airport users is a difficult task for air traffic controllers. One possible way to achieve fair treatment is to introduce a fairness metric into the objective function to be minimized. A straightforward metric is the number of position shifts in the sequence that a flight will accept between pushback (PB) and actual takeoff (TO). In order to keep the objective function purely delay-based, this metric is introduced as a constraint and not as part of the objective function. The "fairness" constraint is introduced through the use of a *"Maximum takeoff Position Shifting"* (MPS) constraint that limits the deviation from a "First Come (Call Ready for Pushback) First Serve (Release to Take Off)" policy, unless specific agreements (known to the optimization planning tool) exist between ATC and the airlines. The MPS value may be predetermined by ATC or by the airlines. The reference position in the FCFS sequence may be based on the scheduled or actual "call ready" or pushback time. The MPS value then determines the range of acceptable takeoff sequence positions for each departure. If X_{PBi} is the pushback sequence position of aircraft i and X_{TOi} is its takeoff sequence position, the MPS value is used in the following constraint:

$$\left|X_{PB_i}-X_{TO_i}\right| \le MPS \Leftrightarrow \begin{cases} X_{PB_i}-X_{TO_i} \le MPS \\ -X_{PB_i}+X_{TO_i} \le MPS \end{cases}$$

$$\Leftrightarrow \begin{cases} -X_{TO_i} \le MPS - X_{PB_i} \\ X_{TO_i} \le MPS + X_{PB_i} \end{cases} \forall\ \text{aircraft}\ i$$

where MPS and X_{PBi} are constants that are known in advance. The takeoff position X_{TOi} can be written as a function of the decision variables X_{ij} as follows: $X_{TO_i} = \sum_{j=1}^{N_S} j * X_{ij}$ and therefore the above constraints become:

$$\begin{cases} -\sum_{j=1}^{N_S} j * X_{ij} \le MPS - X_{PB_i} \\ \sum_{j=1}^{N_S} j * X_{ij} \le MPS + X_{PB_i} \end{cases} \forall\ \text{aircraft}\ i.$$

Various types of *ATC operational constraints* may restrict the sequence position and time that an aircraft can be released for takeoff. For example, an Expected Departure Clearance Time (EDCT) or a Departure Sequencing Program (DSP) restriction is a time-based constraint. Assuming that with the expert input of air traffic controllers a heuristic methodology can be inferred to translate a takeoff time window to a takeoff slot window, ECDT and DSP constraints can be formulated in the following form:

$$\mathrm{EDCT}_{i_1} \leq \sum_{j=1}^{N_S} j * X_{ij} \leq \mathrm{EDCT}_{i_2} \quad \text{or}$$

$$\mathrm{DSP}_{i_1} \leq \sum_{j=1}^{N_S} j * X_{ij} \leq \mathrm{DSP}_{i_2}$$

where EDCT_{i_1}, EDCT_{i_2}, DSP_{i_1} and DSP_{i_2} are the takeoff slot end values, as defined by ATC for flight *i*, that define the EDCT takeoff slot window (typically a 15-minute time window [16]) or the DSP takeoff slot window (typically a 3-minute time window [16]).

The most frequently used ATC operational constraints are Miles In Trail (MIT) and (less frequently) Minutes In Trail (MinT) constraints that impose aircraft separations en route. These can be stated in terms of a minimum required takeoff sequence position separation ΔX_{ik} between flights *i* and *j*, which have an In-Trail restriction, imposed on them:

$$\left|\sum_{j=1}^{N_S} j * X_{ij} - \sum_{j=1}^{N_S} j * X_{kj}\right| \geq \Delta X_{ik} \Leftrightarrow \left|\sum_{j=1}^{N_S} j * (X_{ij} - X_{kj})\right| \geq \Delta X_{ik}$$

$$\Leftrightarrow \begin{cases} \sum_{j=1}^{N_S} j * (X_{ij} - X_{kj}) \geq \Delta X_{ik} \\ \sum_{j=1}^{N_S} j * (-X_{ij} + X_{kj}) \geq \Delta X_{ik} \end{cases}$$

This means that aircraft *i* and *j* must take off at least ΔX_{ij} takeoff slots apart from each other to ensure that the In-Trail separation is not violated when they become airborne.

Lifeguard flights or other type of *priority constraints* can also be modeled in the form of an upper bound $\mathrm{X}_{\mathrm{maxTO}_i}$ on the takeoff sequence position:

$$\sum_{j=1}^{N_S} j * X_{ij} \leq \mathrm{X}_{\mathrm{maxTO}_i}$$

or in terms of inequality constraints between different flights:

$$\sum_{j=1}^{N_S} j * X_{ij} \leq \sum_{j=1}^{N_S} j * X_{kj} \Leftrightarrow \sum_{j=1}^{N_S} j * (X_{ij} - X_{kj}) \leq 0 .$$

At many airports, localized *sequencing constraints* also affect the departure efficiency. For example, back-to-back departures to the same departure fix are generally not allowed because they require additional gaps between flights. Typically these gaps are achieved by alternating jet and propeller aircraft departures on the same runway, because these two different types of aircraft usually use different departure fixes after takeoff. Such constraints can also be introduced in the form of a position constraint.

While the optimization toolbox in Matlab has functions to solve linear programs, there are no functions to solve pure integer programs (in which all the decisions variables can only assume integer values). However, a use-developed function to solve integer programs was downloaded from the file exchange website for Matlab users [20] and was implemented. This function utilizes the optimization functions that are native to Matlab to perform a basic Branch & Bound algorithm implementation. The Branch & Bound algorithm offers a solution method for integer programming or mixed integer programming problems based on an implicit enumerative evaluation of all feasible solutions. The basic principle of the method is to relax the constraint that the variables must be integers, obtaining what is known as the *linear relaxation* of the original problem. The latter is solved with linear programming methods, and that solution is then used to iteratively fix the values of the integer variables in a tree of sub-problems that terminates with the desired optimal integer solution.

4. Preliminary Results

While the Matlab model is not yet complete, it has sufficient functionality to evaluate the fundamental behavior of both stages in the optimization algorithm.

In the first stage of the process, departure class schedules are generated and the throughput for each class schedule is calculated as the total time (in seconds) that it takes to complete all the departure operations in the schedule. Figure 6 shows a sample of the entire set of class schedules developed for nine (9) departing aircraft by the random generation process of stage 1, before crossings are included.

Schedule										Throughput
Index No										Without X
30	L	L	H	L	L	L	S	S	H	540
45	L	L	H	S	S	L	L	L	H	540
63	L	L	L	H	S	S	L	L	H	540
90	L	L	L	H	L	L	S	S	H	540
105	L	L	L	S	S	H	L	L	H	540
3	L	L	H	L	S	S	L	L	H	540
22	L	L	H	L	L	S	S	L	H	540
41	L	L	H	S	L	L	L	S	H	540
71	L	L	L	H	S	L	L	S	H	540
82	L	L	L	H	L	S	S	L	H	540

Figure 6: Departure class schedule without crossings

Figure 7 shows the same set of class schedules as in Figure 6 after they have been "enriched" with the crossing associated with five (5) arrivals during the planning period. The new throughput values for each class schedule includes the time required for crossings. The effect of crossings on throughput is apparent, but, in addition, it can be seen that the same set of crossings changes the throughput of certain schedules less than others, especially when a crossing group is scheduled right after a heavy departure, such as in schedules 3, 22 and 41.

Schedule															Throughput	
Index No															Without X	With X
30	L	L	s	s	h	H	s	l	L	L	L	S	S	H	540	590
45	L	L	s	s	h	H	s	l	S	S	L	L	L	H	540	590
63	L	L	s	s	h	L	s	l	H	S	S	L	L	H	540	630
90	L	L	s	s	h	L	s	l	H	L	L	S	S	H	540	630
105	L	L	s	s	h	L	s	l	S	S	H	L	L	H	540	630
3	L	L	s	s	h	H	s	l	L	S	S	L	L	H	540	590
22	L	L	s	s	h	H	s	l	L	L	S	S	L	H	540	590
41	L	L	s	s	h	H	s	l	S	L	L	L	S	H	540	590
71	L	L	s	s	h	L	s	l	H	S	L	L	S	H	540	630
82	L	L	s	s	h	L	s	l	H	L	S	S	L	H	540	630

Figure 7: Departure class schedule with crossings

After the schedules are ordered by throughput, the best schedules are selected as the set of Target Class Schedules.

The schedule of arrivals and departures used as input to the model was derived from the arrival and departure data collected during observations at the Boston Logan airport control tower on February 2, 1999.

For a Target Class Schedule of nine (9) aircraft, consisting of four (4) small and five (5) large aircraft, the second stage of the optimization produces a final schedule of the form (not all aircraft displayed):

Class Slot Sequence: S s s s S S s S L L L L L
At time 06:33 departure AAL1317 (S 1) takes off in slot 1 with a maximum delay of 1 minute
At time 06:34 the following arrival(s) cross together:
-S arrival N65MJ (number 1 in the arrival schedule) with a delay of 113 seconds
-S arrival USC422 (number 2 in the arrival schedule) with a delay of 10 seconds
-S arrival FDX1464 (number 3 in the arrival schedule) with a delay of 0 seconds
At time 06:38 departure N180M (S 5) takes off in slot 2 with a maximum delay of 1 minute
At time 06:39 departure USC168 (S 4) takes off in slot 3 with a maximum delay of 2 minutes
At time 06:40 the following arrival(s) cross together:
-S arrival ASH5268 (number 4 in the arrival schedule) with a delay of 109 seconds
At time 06:41 departure N109FX (S 3) takes off in slot 4 with a maximum delay of 3 minutes
At time 06:42 departure SGR501 (L 9) takes off in slot 5 with a maximum delay of 2 minutes
At time 06:43 departure USA1854 (L 7) takes off in slot 6 with a maximum delay of 3 minutes
At time 06:44 departure USS6171 (L 8) takes off in slot 7 with a maximum delay of 4 minutes
At time 06:45 departure COA339 (L 6) takes off in slot 8 with a maximum delay of 5 minutes
At time 06:46 departure DAL1821 (L 2) takes off in slot 9 with a maximum delay of 9 minutesTotal **time to complete departures** is 13 minutes
Total departure **delay** is AT MAXIMUM 28 minutes.

The output can be read as follows: At 6:33, Departing flight AAL1317, which is operated by a small aircraft and was number 1 in the pushback sequence, is scheduled to take off in the first departure slot of this schedule. If this happens, AAL1317 will undergo a delay of one minute at most, which means that in the worst case, it is expected to be cleared for take off at most one minute later than the earliest time it could have arrived at the runway queue.

It was assumed that:

- The taxiway space between the two parallel runways of this configuration can accommodate three small, or two large or one heavy aircraft.
- The maximum allowable delay for a crossing aircraft is 150 seconds.
- All small arrivals must exit the runway early and occupy only the first cross-point, while all large and heavy arrivals roll on to the second runway cross-point. This is an assumption that can be relaxed. For example, additional small aircraft can be sent to the second cross-point in order to defer crossings for a while, if there is departure pressure on the runway and if the maximum crossing delay constraint is not going to be violated. This relaxation is actually something to

be added in the model and tested in the future.

Under these assumptions, the crossing part of the output can be read as follows: At 6:34, three small aircraft are scheduled to cross the runway. Their corresponding positions in the touchdown sequence and the crossing delay each has to suffer are also listed. Arriving flights N65MJ, USC422 and FDX1464 are all operated by small aircraft. Based on their expected touchdown times and depending on the time the departing aircraft were expected to reach the runway, the three crossings were scheduled right after the first departure. Even though there was a second departure behind the first one available for takeoff at the runway, the three crossings had already accumulated between the two runways and there was no taxiway space for more arrivals. Therefore, the crossing had to be serviced on the departure runway before more departures were allowed. If priority had been given to a second departure, either the arrival schedule would have to be disrupted due to the lack of taxiway space for additional arrivals to be held while the second departure was being serviced, or the maximum crossing delay constraint would have to be violated.

The output presented above corresponds to the case where no MPS constraints were imposed. The output for the same set of aircraft with MPS = 3, is:

Class Slot Sequence: S s s s S S s S L L L L L
At time 06:33 departure AAL1317 (S 1) takes off in slot 1 with a maximum delay of 1 minute
At time 06:34 the following arrival(s) cross together:
-S arrival N65MJ (number 1 in the arrival schedule) with a delay of 113 seconds
-S arrival USC422 (number 2 in the arrival schedule) with a delay of 10 seconds
-S arrival FDX1464 (number 3 in the arrival schedule) with a delay of 0 seconds
At time 06:38 departure N180M (S 5) takes off in slot 2 with a maximum delay of 1 minute
At time 06:39 departure USC168 (S 4) takes off in slot 3 with a maximum delay of 2 minutes
At time 06:40 the following arrival(s) cross together:
-S arrival ASH5268 (number 4 in the arrival schedule) with a delay of 109 seconds
At time 06:41 departure N109FX (S 3) takes off in slot 4 with a maximum delay of 3 minutes
At time 06:46 departure DAL1821 (L 2) takes off in slot 5 with a maximum delay of 5 minutesAt time 06:42 departure SGR501 (L 9) takes off in slot 6 with a maximum delay of 3 minutes
At time 06:43 departure USA1854 (L 7) takes off in slot 7 with a maximum delay of 4 minutes
At time 06:44 departure USS6171 (L 8) takes off in slot 8 with a maximum delay of 5 minutes
At time 06:45 departure COA339 (L 6) takes off in slot 9 with a maximum delay of 6 minutes
Total **time to complete departures** is 13 minutes
Total departure **delay** is AT MAXIMUM 28 minutes.

Even though total departure delay and runway throughput remain the same for both cases (no MPS and MPS = 3), the effect of the MPS constraint is evident in the slot allocation of the flights at the end of the schedule.

A schedule was also developed for seven (7) departing aircraft. The slot allocation and delay distribution for the 7 aircraft case was then compared to the slot allocation and delay distribution for the 9 aircraft case. As Figure 8 and Figure 9 show, departure slot assignments and therefore delay distribution (i.e. which aircraft will be delayed and for how long) depends on the weight class composition of the Target Class Schedule. For example, in the case of the seven-aircraft schedule, all aircraft change position (for the MPS cases displayed in Figure 8). In the nine-aircraft schedule, only the five large aircraft change position (across the MPS cases displayed in Figure 9).

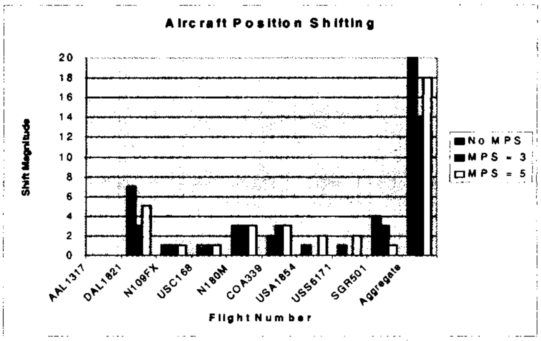

Figure 8: Aircraft position shifting (7 aircraft)

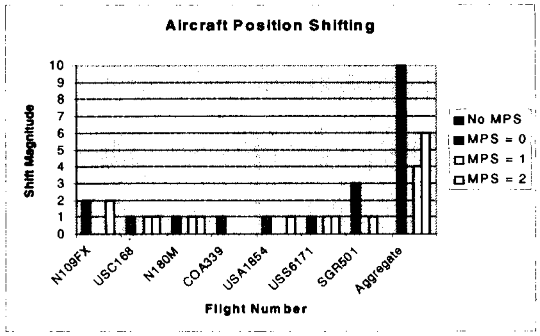

Figure 9: Aircraft position shifting (9 aircraft)

By looking at the aggregate (total among all departures) number of position shifts between pushback and takeoff in both schedule cases, it is clear that the introduction of the MPS constraint, has the desired effect of introducing a certain level of fairness into the final schedule, and that this level of fairness increases as the MPS value decreases.

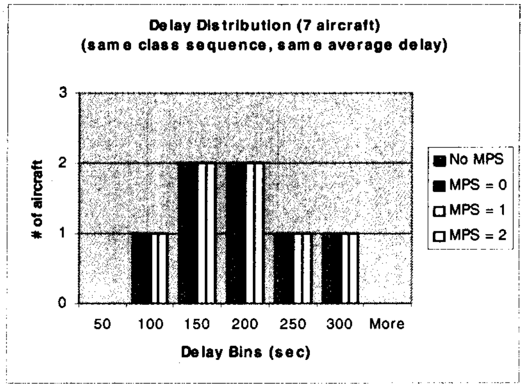

Figure 10: Delay distribution (7 aircraft)

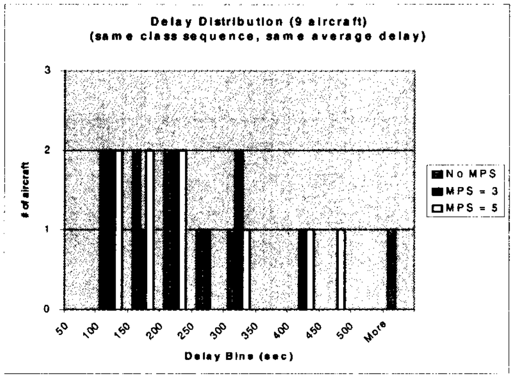

Figure 11: Delay distribution (9 aircraft)

It is also possible that the MPS constraint affects the distribution of delays i.e. the delay penalty applied to specific flights. While the delay distribution remains unchanged in the seven-aircraft case, as shown in Figure 10, in the nine-aircraft case certain flights are penalized significantly when the MPS constraint is active, as shown in Figure 11.

5. *Summary & Future Work*

This paper introduced a "two stage" optimization algorithm for solving the Runway Operations Planning (ROP) problem i.e. to determine the optimal departure schedule.

The sole objective of the first stage is to determine the best departure class sequence (from a throughput perspective) to be used in the second stage. This is achieved by calculating the throughput for each class sequence in a family of "*enriched*" class sequences. The term "enriched" is used to denote the fact that the departure class sequences also include "place holders" for aircraft crossing the departure runway, as well as for arriving aircraft if dual operations are performed on the runway.

The second stage of the optimization algorithm is formulated as an integer program that generates a solution that represents the assignment of aircraft to class slots. Given that throughput maximization is addressed in the first stage of the algorithm, a delay-based objective function is used to address the remaining constraints. Fairness and ATC considerations are introduced into the formulation as constraints.

At the most fundamental level, both stages perform the two functions required to determine the optimal sequence. The class of each departure slot is defined in the first stage, and specific aircraft are assigned to each of the defined class slots in the second stage. While the second stage may be performed immediately after the first, the two stages may also be performed separately depending on the needs of the particular real-world situation.

The "two-stage" algorithm was implemented using Matlab and Simulink. The rationale for this choice is that Matlab offers a robust programming environment, which also supports data structures, and the combination of Matlab and Simulink offers rapid prototyping capabilities and modularity. Thus, it is easy to make changes to the airport surface and runway geometry being modeled.

While the Matlab model is not yet complete, it has sufficient functionality to evaluate the fundamental behavior of both stages in the optimization algorithm. Results indicate that the introduction of the MPS constraint, has the desired effect of introducing a certain level of fairness into the final schedule, and that this level of fairness increases as the MPS value decreases. Results also indicate that the MPS constraint affects the distribution of delays i.e. the delay penalty applied to specific flights.

Apart from the obvious future work of modeling other airport geometries and exploring issues associated with executing the runway operations plans that are developed, there are several model parameters worth exploring. These include the:

- Length of the planning window and the resulting number of aircraft (departures and arrivals) included in the planning window.
- Crossing point (taxiway) capacity and maximum crossing delay constraints. They affect the location and length of the crossing "gaps" that must be injected into the departure schedule.
- Probability distributions for the pushback and taxi processes. They affect the stochastic

throughput calculations of class schedules the fidelity of the model vis a vi the real world.

6. *Acknowledgements*

This research was supported in part by NASA grant NAG 2-1128. The authors would like to thank the staff of the Boston Tower and TRACON for their help in arranging field observations.

References

1. Anagnostakis, I., Clarke, J-P., Böhme, D., Völckers, U., "Runway Operations Planning and Control: Sequencing and Scheduling," *AIAA Journal of Aircraft*, November-December 2001, Vol. 38, No.6, also presented at the *34th Hawaii International Conference on System Sciences*, Maui, Hawaii, January 3rd – 6th, 2001
2. Anagnostakis, I., Idris, H. R., Clarke, J-P., Feron, E., Hansman, R.J., Odoni, A.R., Hall, W.D., "A Conceptual Design of a Departure Planner Decision-Aid," International Center for Air Transportation, MIT, *AIAA Progress in Aeronautics and Astronautics: Air Transportation Systems Engineering*, American Institute of Aeronautics and Astronautics, Vol. 193, Chapter 21, Washington, District of Columbia, also at the *3rd USA/Europe International Air Traffic Management R & D seminar*, ATM-2000, proceedings, session E: Airport, Ground Operations, Arrival and Departure Management, Naples, Italy, 13-16 June 2000, sponsored by FAA & Eurocontrol, available at http://atm-seminar-2000.eurocontrol.fr/ [cited Oct. 6, 2000]
3. Anagnostakis, I., Carr, F., Evans, A., Idris, H., "Preliminary Observations at Washington Dulles Airport (IAD), May 2001", Massachusetts Institute of Technology, 2001, available at http://icat-server.mit.edu/~fcarr/Publications/IAD_Prelim.ppt. [cited May. 22, 2002]
4. Andersson, K., Carr, F., Feron, E., Hall, W., "Analysis and Modeling of Ground Operations at Hub Airports", *presented at the 3rd USA/Europe Air Traffic Management R&D Seminar*, Naples, Italy, 13rd - 16th June 2000
5. Atkins, S. and Hall, W., "A Case for Integrating the CTAS Traffic Management Advisor and the Surface Management System", *AIAA Guidance, Navigation, and Control Conference*, Denver, CO, Aug. 14-17, 2000
6. Atkins, S. and Brinton, C., "Concept Description and Development Plan for the Surface Management System", *Journal of Air Traffic Control, Vol. 44, No. 1, January – March, 2002*
7. Carr, F., Evans, A., Clarke, J-P., Feron, E., "QMAST: Queuing Model for Airport Surface Traffic", Massachusetts Institute of Technology, 2002, available at http://icat-server.mit.edu/~fcarr/online_model/ [cited April 2002]
8. Carr, F., Evans, A., Clarke, J-P., Feron, E., "Modeling and Control of Airport Queuing Dynamics under Severe Flow Restrictions." *American Control Conference 2002 proceedings (ISBN 0-7803-7299-9)*, Anchorage, Alaska, May 8-10, 2002
9. Cooper, W.W., Mohleji, S.C., Burke, C.D., Foster, G., Mills, M.J., "Determination of Minimum Push-Back and Taxi-Out Time Predictability Needed For Near-Term Departure Scheduling using DEPARTS" *presented at the 4th USA/Europe Air Traffic Management R&D Seminar*, Santa Fe, New Mexico, USA, 3rd - 7th December 2001
10. DEPARTS research project summary can be found at: http://www.mitre.org/technology/mtp01/quads_charts/decision_support/cooper.shtml
11. Glass B.J., Gawdiak, Y., "Integrated Navigation Applications in Airport Surface Traffic Management," *4th International Conference on Integrated Navigation Systems*, St. Petersburg, Russia, May 1997
12. Glass B.J., "Automated Data Exchange and Fusion for Airport Surface Traffic Management," AIAA 97-3679, *proceedings of the AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, Aug. 11-13, 1997, pp. 904 - 911
13. Idris, H.R., Delcaire, B., Anagnostakis, I., Hall, W.D., Hansman, R.J., Feron, E., Clarke, J-P., Odoni, A., "Observations of Departure Processes at Logan Airport to Support the Development of Departure Planning Tools," 2nd USA/Europe International Air Traffic Management R & D seminar, ATM-98, Orlando, FL, 1 – 4 December 1998
14. Idris, H.R., Anagnostakis, I., Delcaire, B., Hansman, R.J., Clarke, J-P., Feron, E., Odoni, A.R., "Observations of Departure Processes at Logan Airport to Support the Development of Departure Planning Tools," *Air Traffic Control Quarterly Journal*, Special Issue in Air Traffic Management, Volume 7, Number 4, 1999, pp. 229-257
15. Idris, H.R., Delcaire, B., Anagnostakis, I., Hall, W.D., Pujet, N., Feron, E., Hansman, R.J., Clarke, J-P., and Odoni, A.R., "Identification of Flow Constraint and Control Points in Departure

Operations at Airport Systems," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, Aug. 1998, Vol.2, pp.947-956

16. Idris, H.R., "Observation and Analysis of Departure Operations at Boston Logan International Airport," *PhD. Thesis*, Department of Mechanical Engineering, Massachusetts Institute of Technology, September 2000.
17. Idris, H.R., Clarke, J-P., Bhuva, R.R., Kang, L.S., Queuing Model for Taxi-Out Time Estimation, *Air Traffic Control Quarterly Journal*, Volume 10, Number 1, 2002
18. Lawson, D. R., "Surface Movement Advisor (SMA)", Journal of Air Traffic Control, January-March, 1998
19. Pujet, N., Feron, E., "Input-Output Modeling-and Control of the Departure Process of Congested Airports", Air Traffic Control Quarterly Journal, Volume 8, Number 1, pp. 1 – 32, 2000
20. Solberg, I., "fminconset: solves constrained minimization problems, when some of the variables are restricted to discrete values", Matlab File Exchange, http://www.mathworks.com/matlabcentral/fileexchange/index.jsp
21. Surface Management Systems (SMS) research summaries at:
    a) http://ffc.arc.nasa.gov/our_projects/sms.html
    b) http://www.metronaviation.com/airport-surface-mgt.html