AIAA Guidance, Navigation, and Control Conference
10 - 13 August 2009, Chicago, Illinois

AIAA 2009-6250

A Generalized Dynamic Programming Approach for a Departure Scheduling Problem

Sivakumar Rathinam*

Texas A & M University, College Station, 77840

Zachary Wood†

University of California-Santa Cruz Affliated Research Center, Moffett Field, CA, 94035

Banavar Sridhar‡ and Yoon Jung§

NASA Ames Research Center, Moffett Field, CA, 94035

Reducing the delays of the departing aircraft can potentially lead to improving the efficiency of the surface operations at airports. This paper addresses a departure scheduling problem with an objective to reduce total aircraft delays subject to timing and ordering constraints. The ordering constraints model the queuing area of airports where the aircraft align themselves in the form of chains before departing. By exploiting the structure of the problem, a generalized dynamic programming approach is presented to solve the departure scheduling problem optimally. Computational results indicate that the approach presented in this paper is reasonably fast, *i.e.*, it takes less than one tenth of a second on average to solve a 40 aircraft problem. Also, the approach produces optimal sequences whose delay is approximately 12 minutes, on average, less than the delays produced by the First Come First Serve (FCFS) sequences.

Nomenclature

$\mathcal{A}$	Number of aircraft
$\alpha(i)$	Release time of aircraft i
$t(i)$	Departure time of aircraft i (decision variable)
$sep(i,j)$	Minimum separation in time required for aircraft j to wait to depart after aircraft i
l	Number of queues
q_i	Number of aircraft in queue i
m	Variable representing the queue
a^i_j	j^{th} aircraft waiting to depart in the i^{th} queue
k_i	Total number of aircraft departed from the i^{th} queue
$(m, k_1, \ldots, k_l)$	A state defined by all the departed aircraft given that the last departing aircraft is from queue m
$\mathcal{S}$	Set of all possible states
$SPAN(m, k_1, \ldots, k_l)$	Optimum throughput corresponding to the state $(m, k_1, \ldots, k_l)$
$DELAY(m, k_1, \ldots, k_l)$	Optimum delay corresponding to the state $(m, k_1, \ldots, k_l)$
$DELAY_s(m, k_1, \ldots, k_l)$	Delay of a sequence, s, corresponding to the state $(m, k_1, \ldots, k_l)$
$LAST_s(m, k_1, \ldots, k_l)$	Departure time of the last departed aircraft of a sequence s corresponding to the state $(m, k_1, \ldots, k_l)$
$\mathcal{F}_{nd}(m, k_1, \ldots, k_l)$	Set of all the non-dominated feasible sequences

*Assistant Professor, Mechanical Engineering, Texas A & M University, College Station, TX 77840. srathinam@tamu.edu.

†Development Engineer, University of California-Santa Cruz Affliated Research Center, Moffett Field, CA 94035. zachary.p.wood@nasa.gov.

‡Senior Scientist, NASA Ames Research Center, Moffett Field, CA 94035. Banavar.Sridhar@nasa.gov. AIAA fellow.

§Aerospace Engineer, NASA Ames Research Center, Moffett Field, CA 94035. yoon.c.jung@nasa.gov. AIAA member.

corresponding to the state $(m, k_1, \dots, k_l)$.

Subscript

i Variable number

I. Introduction

Reducing delays of aircraft operating at any airport is becoming important as traffic demand and congestion increases. The problem addressed in this paper is that of a departing scheduling problem that arises on the surface of an airport. The objective of the Departure Scheduling Problem (DSP) is to find a departure time for each aircraft such that the total delay of all the aircraft is minimized subject to timing, separation and ordering constraints. The delay for an aircraft is defined as the difference between its departure time and its release time (*i.e.*, earliest possible departure time). The timing constraint requires that the departure time assigned to an aircraft is at least equal to its release time. Due to the wake vortex generated by departing aircraft, the separation constraint requires that the departure times of any two aircraft must be separated at least by a constant that is dependent on the type of the two departing aircraft. It is assumed that the ordering constraints are in a form of chains as illustrated in Fig. 1. These chain like ordering constraints represent a simplified model of the runway queue structure present at airports such as the Dallas Fort Worth International (DFW) Airport.

In this paper, a generalized dynamic programming approach discussed in Carraway and Morin[1] is used to solve the DSP optimally. This approach improves on the work by Psaraftis[15] . Psaraftis proposed a dynamic programming approach to address a DSP with no timing constraints that restrict the departure time of each aircraft to be at least greater than its release time. Computational results indicate that the approach proposed in this paper is reasonably fast, *i.e.*, it takes less than one tenth of a second on an average to solve a 40 aircraft DSP with three queues. Also, the approach produces optimal sequences whose delay is approximately 12 minutes, on an average, less than the delays produced by the First Come First Serve (FCFS) sequences.

The DSP is formulated in Section II and a review of the available literature is presented in Section III. In Section IV, it is shown why a direct extension of the dynamic programming approach by Psaraftis may not work for the DSP. The generalized dynamic programming approach for the DSP is presented in Section V. The approach presented in this paper can also be extended to some generalizations of the DSP as shown in Section VII. The paper ends with conclusions in Section VIII.

II. Problem Formulation

Consider a set of n departing aircraft denoted by $\mathcal{A} = \{1, 2, \dots, n\}$. Let $t(i)$ be the decision variable that denotes the departure time for the i^{th} aircraft. Aircraft i is available to depart only after its release time which is denoted by $\alpha(i)$ (*i.e.*, $t(i) \geq \alpha(i)$). If aircraft i departs before aircraft j, then their corresponding departure times (*i.e.*, $t(i)$ and $t(j)$) must be at least separated by a constant denoted by $sep(i, j)$. This separation requirement depends on whether aircraft i departs before aircraft j or vice versa (*i.e.*, $sep(i, j)$ need not be equal to $sep(j, i)$). There are l queues available and each aircraft must be present in one of those queues. It is assumed that the separation times satisfy the triangle inequality, *i.e.* $sep(i, j) + sep(j, k) \geq sep(i, k)$ $\forall i, j, k, i \neq j \neq k$. Let the order of aircraft present in the i^{th} queue be denoted by $\{a^i_1, \dots, a^i_{q_i}\}$ where q_i be the total number of aircraft present in the i^{th} queue. In the given order for the i^{th} queue, aircraft a^i_1 must depart before a^i_2, a^i_2 must depart before a^i_3 and so on. The delay of the i^{th} aircraft is defined as $t(i) - \alpha(i)$. The objective of the scheduling problem is to determine the departure times of all the aircraft that minimizes the total delay, $\sum_{i=1}^{n} (t(i) - \alpha(i))$, subject to the ordering and timing constraints.

III. Background and Literature Review

There are several heuristics[3−6] and exact algorithms[10,11,15,18,19] available for addressing aircraft scheduling problems in the literature. Most of the work related to the DSP in air traffic control has been in the area of scheduling aircraft landings. The constraints in problems involving landing aircraft are very similar to that of the constraints in the DSP. The precedence or ordering constraints of the DSP addressed in this paper has a special structure where the departing aircraft are queued up in the form of chains. These

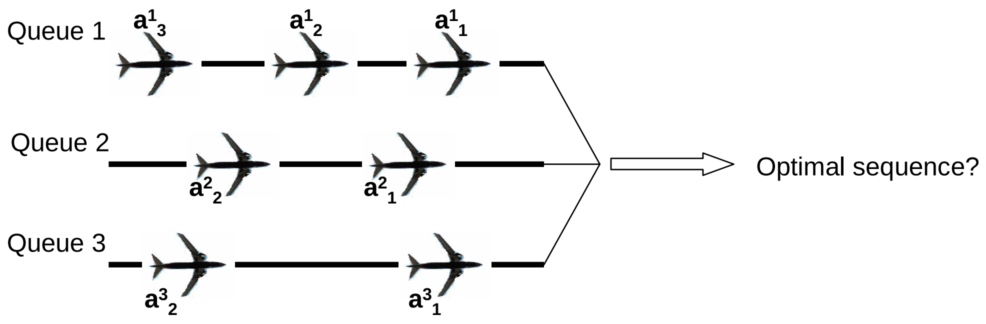


Figure 1. The DSP with three queues. Aircraft cannot overtake each other if they belong to the same queue.

chain like ordering constraints present in the DSP represent a simplified model of the physical layout of the runway queues present in airports such as the DFW airport. A landing aircraft problem may not have this special structure.

Irrespective of whether an algorithm produces an optimal or a good, approximate solution, it is important to note that it would be useful to develop algorithms that can ultimately be used in a real-time simulation system. An exact algorithm produces optimal solutions but may have a running time that could make it less suitable in a real-time simulation. On the other hand, a heuristic could run fast but there is no guarantee on the quality of the solutions it produces. Optimal costs or tight lower bounds to the optimal costs are anyway required to evaluate the quality of a heuristic. Currently, the authors are involved in the development of the Surface Management System[2] (SMS) that can provide real-time advisories to human controllers on scheduling movements of aircraft at airports. In this context, one of the goals of the paper was to develop an algorithm that can produce high quality solutions for the DSP involving 40 aircraft in the order of seconds. The motivation behind choosing a 40 aircraft instance is that during peak hours at DFW airport, there are approximately 40 departures in one hour from each of the main runways. Due to the uncertainties involved in the release times of the aircraft, algorithms are expected to plan departure schedules for at most an hour. In the following discussion, a review of the existing literature related to the single runway, aircraft scheduling problem is presented.

Dear and Sherif[3,4] were among the earliest to address the static and dynamic scheduling of landing aircraft. In static scheduling, a sequence/schedule is determined for a given set of aircraft. In dynamic scheduling, new aircraft are added continuously to the system and the schedules need to be updated frequently to include the new set of aircraft. Dear and Sherif[3] introduced the concept of Constrained Position Shifting (CPS) as a feasible way to address the dynamic problem. In this concept, a First Come First Served Sequence (FCFS) is initially generated based on the predicted landing times of all the aircraft. Then, an optimal sequence is generated such that no aircraft can be shifted more than a given number of positions away from its original position in the FCFS sequence. For example, if the position of an aircraft in the FCFS sequence is 5 and the maximum number of shifts allowed is one, then the aircraft in the optimal sequence can be in positions 4, 5 or 6. If CPS is not present, the position of an aircraft can be shifted several places for each update of the aircraft sequence. Therefore, by incorporating CPS while scheduling aircraft, one can eliminate these huge shifts in the positions of the aircraft. Heuristics were presented in Dear and Sherif[4] to solve the aircraft scheduling problem with CPS. In Section VII, it is shown how the approach presented in this paper can also be extended to find optimal solutions for the DSP with the CPS constraints.

There are several other heuristics available for variants of the aircraft scheduling problems[5−9,11] . Venkatarishnan et al.[5] presented a heuristic based on the dynamic programming approach by Psaraftis to solve the arrival scheduling problem with time window constraints. Genetic algorithms are given in Abela et al.[6] , and Ernst et al.[11] to solve a generalization of the arrival scheduling problem. Meta-heuristics including simulated annealing and tabu search methods are presented for a generalization of the DSP in Atkin et al.[7−9] where the authors are motivated by the taxi layout of the London Heathrow Airport with complex holding point structures and additional constraints.

There are few ways in which optimal solutions can be obtained for aircraft scheduling problems. One way is to formulate the problem as an integer or a mixed integer linear program[10, 11, 17] and solve the resulting program using any standard optimization software like CPLEX. This approach has a drawback, in the sense that the running times of the solvers could vary significantly[17] depending on a given instance of the problem. However, it is important to note that this approach can deal with several generalizations of the DSP. For example, it can readily deal with the problems where the assumption on the separation times for the aircraft (based on the previously mentioned triangle inequality) is not satisfied.

Another method to find optimal solutions for aircraft scheduling problems is to use the dynamic programming[15−19] as explained in the following section. This approach exploits the chain-like ordering of the aircraft. The next section also serves as a motivation for the approach used in this paper.

IV. Dynamic Programming

To instigate the use of generalized dynamic programming for solving the DSP, the approach given by Psaraftis[15] for a similar aircraft scheduling problem is first presented. Later, it is shown that the direct application of the dynamic programming approach used by Psarftis[15] may not work for the DSP if the objective is to minimize the total delay of all the aircraft and there are timing constraints for each aircraft as present in the DSP.

The approach in Psaraftis[15] can be used to solve a similar sequencing problem called the Makespan Scheduling Problem (MSP) with an objective of minimizing the makespan (*i.e.*, the departure time of the last aircraft given by $\max_{i=1}^{n}(t_i)$) subject to exactly the same constraints as in the DSP. Minimizing the departure time of the last aircraft is also important in air traffic applications as this objective corresponds to maximizing the runway throughput.

The MSP is associated with a set of states, $\mathcal{S}$, where each state $(m, k_1, \ldots, k_l) \in \mathcal{S}$ is defined by the number of aircraft that has departed from each of the queues (*i.e.*, k_i aircraft has departed from the i^{th} queue for $i = 1, \ldots, l$) given that the last departing aircraft is from queue m. Let the value function, $SPAN(m, k_1, \ldots, k_l)$, denote the optimal makespan corresponding to the state $(m, k_1, \ldots, k_l)$.

This value function can be computed recursively using the following equations:

$$SPAN(m, k_1, \ldots, k_l) \quad = \quad \begin{cases} 0, & \text{if } k_1 = k_2 = \ldots = k_l = 0, \\ \\ \min_{n \in Q} \max\left(SPAN(n, k_1', \ldots, k_l') + sep(a^n_{k_n'}, a^m_{k_m}), \alpha(a^m_{k_m})\right), & \text{otherwise} \end{cases} \tag{1}$$

where

$$Q = [i : k_i' > 0] \tag{2}$$

and for $i = 1, \ldots, l$,

$$k_i' \quad = \quad \begin{cases} k_i - 1, & \text{if } i = m, \\ k_i, & \text{otherwise.} \end{cases} \tag{3}$$

The optimal makespan for the MSP is then given by $OPT_{msp} = \min_{m=1,\ldots,l} SPAN(m, q_1, \ldots, q_l)$. The method to compute OPT_{msp} is to first start with $SPAN(m, k_1, \ldots, k_l)$ for $k_1 = k_2 = \ldots = k_l = 0$ and then compute $SPAN(m, k_1, \ldots, k_l)$ recursively for lexicographically increasing values of k_1 through k_l. While performing the recursion for the state $(m, k_1, \ldots, k_l)$, the optimal queue corresponding to the minimization of $SPAN(m, k_1, \ldots, k_l)$ can also be stored in an array denoted by $SOL(m, k_1, \ldots, k_l)$. In this way, the optimal sequence corresponding to the states in $\mathcal{S}$ can be retrieved at the end of the recursion. In general, the dynamic programming approach for any problem relies on the following strong principle of optimality:

Any optimal policy has a property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

It is easy to check that the definition of state and the value function given in equations (1-3) ensures that the principle of optimality is satisfied and as a result the equations *correctly* produce an optimal solution for the MSP. Now, let us directly extend the above approach to the objective of the DSP where the total aircraft delays, given by $\sum_{i=1}^{n}(t(i)-\alpha(i))$, must be minimized. For the case when the release times of all the aircraft are zero (*i.e.*, $\alpha_i = 0$ for $i = 1, \ldots, n$), by modifying the definition of the value function but using the same set of states as in $\mathcal{S}$, Psaraftis[15] showed how to compute an optimal DSP sequence. In this paper, the more general case is considered when at least one of the release times of the aircraft is not equal to zero. This general case was not a constraint for the scheduling problem addressed in Psaraftis[15] . However, for this general case, it seems difficult to retain the same definition of state and define a value function that is computable recursively and satisfies the strong principle of optimality. The following discussion attempts to explain this difficulty and will also motivate the use of generalized dynamic programming[1] for the DSP.

Definition IV.1 *Let $DELAY_{opt}(m, k_1, \ldots, k_l)$ denote the total optimal delay corresponding to the state $(m, k_1, \ldots, k_l)$. Also, let $LAST_{opt}(m, k_1, \ldots, k_l)$ denote the makespan (the departure time of last departing aircraft $a^m_{k_m}$) corresponding to the optimal departure sequence of $DELAY_{opt}(m, k_1, \ldots, k_l)$.*

For the following argument, let us assume that there are only two queues ($l = 2$). Also, let us assume that there is a unique departure sequence that minimizes $DELAY_{opt}(1, k_1, k_2)$ and a unique departure sequence that minimizes $DELAY_{opt}(2, k_1, k_2)$. Now, let us try to compute $DELAY_{opt}(1, k_1 + 1, k_2)$ recursively using already computed optimal sequences for $DELAY_{opt}(1, k_1, k_2)$ and $DELAY_{opt}(2, k_1, k_2)$. $DELAY_{opt}(1, k_1 + 1, k_2)$ is the optimal cost corresponding to the state where $k_1 + 1$ and k_2 aircraft have already departed from queue 1 and queue 2, and the last departed aircraft was from queue 1 (*i.e.*, the aircraft denoted by $a^1_{k_1+1}$). As there are only two queues, the aircraft that departed before $a^1_{k_1+1}$ can either be $a^1_{k_1}$ or $a^2_{k_2}$. Therefore the optimal delay cost, $DELAY_{opt}(1, k_1 + 1, k_2)$, and its corresponding makespan, $LAST_{opt}(1, k_1, \ldots, k_l)$, can either be equal to

$$DELAY_{opt}(1, k_1 + 1, k_2) = Cost_1 \text{ and } LAST_{opt}(1, k_1 + 1, k_2) = Time_1,$$

or be equal to

$$DELAY_{opt}(1, k_1 + 1, k_2) = Cost_2 \text{ and } LAST_{opt}(1, k_1 + 1, k_2) = Time_2,$$

where

$$\begin{aligned} Cost_i &= DELAY_{opt}(i, k_1, k_2) + \max(LAST_{opt}(i, k_1, k_2) + sep(a^i_{k_i}, a^1_{k_1+1}), \alpha(a^1_{k_1+1})) - \alpha(a^1_{k_1+1}), \\ Time_i &= \max(LAST_{opt}(i, k_1, k_2) + sep(a^i_{k_i}, a^1_{k_1+1}), \alpha(a^1_{k_1+1})), \end{aligned} \tag{4}$$

for $i = 1, 2$. Consider the case when $Cost_1 < Cost_2$. In this case, $DELAY_{opt}(1, k_1 + 1, k_2) = Cost_1$ and $LAST_{opt}(1, k_1 + 1, k_2) = Time_1$. However, it might still not be possible to ignore the sequence corresponding to $Cost_2$ even though $Cost_2 > Cost_1$. The reason why the solution corresponding to $Cost_2$ cannot be ignored is because if we are interested in computing the next optimal cost, $DELAY_{opt}(1, k_1 + 2, k_2)$, then it is not only dependent on $DELAY_{opt}(1, k_1 + 1, k_2)$, but *also* dependent on the departure time of the last departed aircraft, $LAST_{opt}(1, k_1 + 1, k_2)$. It is possible that $Cost_1$ is less than $Cost_2$ but $Time_1$ might be greater than $Time_2$. In such a scenario, the solution corresponding to $Cost_2$ might lead to a better sequence for $DELAY_{opt}(1, k_1 + 2, k_2)$ even though it was not optimal for $DELAY_{opt}(1, k_1 + 1, k_2)$.

If the principle of optimality is not satisfied by the optimal value function and its corresponding states, then it is possible that the definition of state is not rich enough, *i.e.*, it does not carry enough of the process history to determine the optimality of the remaining decisions. There are few ways to enhance the definition of state to address this challenge. Invariably, any enhancement will significantly increase the size of the state space. If one can extend the state to include time also in the definition of state, then it is possible to use dynamic programming to solve the DSP as shown for similar problems in Bianco et al.,[12] Lee and Balakrishnan.[19] However, this approach requires discretizing time into intervals and hence increases the number of states as a function that is inversely proportional to the interval size. There is also an alternate approach to address this challenge for DSP through generalized dynamic programming[1] without increasing the size of the state space. Theoretically, though generalized dynamic programming also could increase the running time significantly, computational results presented in Section VI suggest that this approach is reasonably fast for the DSP.

V. A Generalized Dynamic Programming Approach

Instead of having just one objective of minimizing total delay, let us assume that there are two objectives where the first objective is to minimize total delay and the second objective is to minimize the departure time of the last departed aircraft. When there are multiple objectives, instead of optimal solutions, one is interested in determining *pareto-optimal*) solutions or policies. Consider two different departure sequences s and s' with their corresponding aircraft departure times being $t_1, t_2, \ldots, t_n$ and $t'_1, t'_2, \ldots, t'_n$ respectively. With respect to the two objectives, the sequences s and s' are non-dominated if

$$\text{either } \sum_{i=1}^{n}(t(i) - \alpha(i)) \geq \sum_{i=1}^{n}(t'(i) - \alpha(i)) \text{ and } \max_{i=1,\ldots,n} t(i) < \max_{i=1,\ldots,n} t'(i)$$

$$\text{or } \sum_{i=1}^{n}(t(i) - \alpha(i)) < \sum_{i=1}^{n}(t'(i) - \alpha(i)) \text{ and } \max_{i=1,\ldots,n} t(i) \geq \max_{i=1,\ldots,n} t'(i).$$

Generalized dynamic programming approach[1] can be used for addressing multi-objective problems and is based on the following weakened principle of optimality:

Any non-dominated policy has a property that whatever the initial state and initial decision are, the remaining decisions must constitute a non-dominated policy with regard to the state resulting from the first decision.

By including both the cost incurred due to total delay and the departure time of the last departure aircraft as the two objectives, the generalized dynamic programming approach will eliminate scenarios such as the one discussed in the previous situation where any two departure sequences corresponding to a state are compared purely based on their total delays. Prior to the application of this approach to the DSP, few definitions are stated first in the following discussion.

Definition V.1 *A feasible departure sequence corresponding to the state $(m, k_1, \ldots, k_l)$ is a sequence where the first k_i aircraft has already departed from the i^{th} queue $(i = 1, \ldots, l)$ and the last departed aircraft is from queue m. Let the first objective, $DELAY_s(m, k_1, \ldots, k_l)$, denote the **total delay** of a feasible sequence, s, corresponding to the state $(m, k_1, \ldots, k_l)$. Let the second objective, $LAST_s(m, k_1, \ldots, k_l)$, denote the departure time of last departing aircraft, $a^m_{k_m}$, of the feasible sequence, s, corresponding to the state $(m, k_1, \ldots, k_l)$.*

Definition V.2 *Departure sequences s, s' corresponding to the state $(m, k_1, \ldots, k_l)$ are considered to be non-dominated*

- *if either $DELAY_s(m, k_1, \ldots, k_l) \leq DELAY_{s'}(m, k_1, \ldots, k_l)$ and $LAST_s(m, k_1, \ldots, k_l) > LAST_{s'}(m, k_1, \ldots, k_l)$,*
- *or $DELAY_s(m, k_1, \ldots, k_l) > DELAY_{s'}(m, k_1, \ldots, k_l)$ and $LAST_s(m, k_1, \ldots, k_l) \leq LAST_{s'}(m, k_1, \ldots, k_l)$.*

Definition V.3 *Let $\mathcal{F}_{nd}(m, k_1, \ldots, k_l)$ be the set of **all** the non-dominated feasible sequences corresponding to the state $(m, k_1, \ldots, k_l)$. All the sequences in the set, $\mathcal{F}_{nd}(m, k_1, \ldots, k_l)$, are non-dominated if any pair of distinct sequences in $\mathcal{F}_{nd}(m, k_1, \ldots, k_l)$ are non-dominated.*

If $\mathcal{F}_{nd}(m, k_1, \ldots, k_l)$ has only one sequence, then that unique sequence must be optimal for both the objectives defined by the total delay and the makespan. If $\mathcal{F}_{nd}(m, k_1, \ldots, k_l)$ has more than one sequence, then there must be at least one sequence in $\mathcal{F}_{nd}(m, k_1, \ldots, k_l)$ that minimizes the total delay cost and an other distinct sequence that must be optimal for the makespan objective. The optimal sequence for the DSP can be found by first computing the set of all the non-dominated sequences corresponding to the states $\mathcal{F}_{nd}(m, k_1, \ldots, k_l)$ for $m = 1, \ldots, l$ and then choosing a sequence that minimizes the total delay cost. $\mathcal{F}_{nd}(m, k_1, \ldots, k_l)$ can be calculated recursively using the following equations:

$$\begin{aligned}\mathcal{F}_{nd}(m, k_1, \ldots, k_l) \;&=\; \{s : s \in \mathcal{F}(m, k_1, \ldots, k_l), \\ &\qquad s \text{ and } s' \text{ are non-dominated for any } s' \in \mathcal{F}(m, k_1, \ldots, k_l), s \neq s'\}, \end{aligned} \tag{5}$$

where,

$$\mathcal{F}(m, k_1, \ldots, k_l) \;=\; \begin{cases} \emptyset, & \text{if } k_1 = k_2 = \ldots = k_l = 0, \\ \\ \{s : s = (r, a^m_{k_m}), r \in \mathcal{F}_{nd}(n, k'_1, \ldots, k'_l), n \in Q\} & \text{otherwise}, \end{cases} \tag{6}$$

and Q, k_i are defined in equations (2,3) respectively. To check whether two sequences s and s' are non-dominated in $\mathcal{F}(m, k_1, \ldots, k_l)$ in equation (5), one needs the values of $LAST_s(m, k_1, \ldots, k_l)$ and $DELAY_s$ $(m, k_1, \ldots, k_l)$ for any sequence $s \in \mathcal{F}(m, k_1, \ldots, k_l)$. Now, note that a sequence s in $\mathcal{F}(m, k_1, \ldots, k_l)$ was formed by adding aircraft $a^m_{k_m}$ to the end of r where r is a sequence that was added in the previous stage (refer to equation 6). When an aircraft $a^m_{k_m}$ is added at the end of a sequence r to form a new sequence s, the departure time of the last aircraft in s and its total delay can be computed as follows:

$$\begin{aligned} LAST_s(m, k_1, \ldots, k_l) \;&=\; \begin{cases} \alpha(a^m_{k_m}), & \text{if } k_1 = k_2 = \ldots = k_l = 0, \\ \\ \max(\alpha(a^m_{k_m}), LAST_r(n, k'_1, \ldots, k'_l) + sep(a^n_{k'_n}, a^m_{k_m})) & \text{otherwise}, \end{cases} \\ DELAY_s(m, k_1, \ldots, k_l) \;&=\; DELAY_r(n, k'_1, \ldots, k'_l) + LAST_s(m, k_1, \ldots, k_l) - \alpha(a^m_{k_m}). \end{aligned} \tag{7}$$

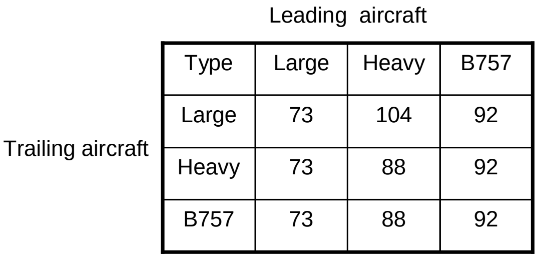
The proof that the above recursive equations *correctly* computes all the non-dominated solutions for each state is straightforward and can be referred to in Carraway and Morin.[1]

VI. Simulation Results

Simulation results are presented in this section to answer the following two important questions: 1) is the generalized dynamic programming approach proposed for solving the DSP fast enough to be considered for implementation in a real-time decision support tool? and 2) on an average, how does the total delay corresponding to an optimal sequence for the DSP compare with the total delay corresponding to a sequence computed from a First Come First Serve (FCFS) discipline? A FCFS discipline just orders the aircraft based on their release times and is generally considered as a baseline solution over which the benefits of optimization are assessed.

The approach presented in the previous section was applied to a DSP with $l = 3$ departure queues. The number of departure queues was chosen to be three because the DSP was motivated by the taxi layout of DFW airport where the main departure runways have three queues. The types of aircraft considered in the simulations were *Large, B757* and *Heavy.* The separation matrix given in Fig. 2 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 73 seconds.

Any departure scheduling algorithm would be used to find the departure times of aircraft over a planning horizon. Generally, this planning horizon may not be more than an hour because of the uncertainties involved



		Leading aircraft		
	Type	Large	Heavy	B757
Trailing aircraft	Large	73	104	92
	Heavy	73	88	92
	B757	73	88	92

Figure 2. Minimum required departure separation between aircraft in seconds.

in predicting the release times of the aircraft. For the study, the planning horizon was varied from ten minutes to one hour in increments of five minutes. The planning horizon determines the number of aircraft used in the study. During peak hours, as there are approximately forty departures in one hour (3600 seconds) from each of the main runways at DFW airport, if T is the planning horizon in seconds, the number of aircraft corresponding to the planning horizon was chosen to be $n_T = \lfloor \frac{T \times 40}{3600} \rfloor = \lfloor \frac{T}{90} \rfloor$. For example, $T = 1800$ corresponds to an instance with 20 departing aircraft. As a result, varying the planning horizon from ten minutes to one hour correspondingly varies the number of aircraft from 6 to 40.

For each time period T (or equivalently the number of aircraft n_T), 100 instances were generated. Currently, most of the departing aircraft at DFW airport are of type *Large*. This scenario could change in the future and is dependent on factors such as traffic demand, fuel costs and airline preferences. For this simulation study, aircraft types were chosen so that no particular type is dominant. If $\{1, \ldots, n_T\}$ denote the aircraft used in a DSP instance, then the first n_l aircraft, $\{1, \ldots, n_l\}$, were chosen to be of type *Large*, the next n_{757} aircraft, $\{n_l + 1, \ldots, n_l + n_{757}\}$, were chosen to be of type *B757* and the remaining n_h aircraft, $\{n_l + n_{757} + 1, \ldots, n_T\}$, were chosen to be of type *Heavy* where,

$$\begin{aligned} n_l &= \lfloor \frac{n_T}{3} \rfloor, \\ n_{757} &= \lfloor \frac{n_T}{3} \rfloor, \\ n_h &= n_T - n_l - n_{757}. \end{aligned} \tag{8}$$

The above numbers were chosen so that the number of aircraft corresponding to any aircraft type is approximately equal to one third of the total number of aircraft. To generate an instance and to assign aircraft to the queues the following rules were used:

- The number of aircraft in each of the queues except the l^{th} queue was chosen to be equal to $\lfloor \frac{n_T}{l} \rfloor$ (*i.e.*, $k_i = \lfloor \frac{n_T}{l} \rfloor$ for $i = 1, \ldots, l-1$).
- The number of aircraft, k_l, in the l^{th} queue was equal to $n_T - (l-1)\lfloor \frac{n_T}{l} \rfloor$.
- The release time of each aircraft (in seconds) was chosen from an uniform distribution on the interval $[0, T]$.
- A random permutation of n_T aircraft was generated with the first k_1 aircraft from the permutation assigned to the 1^{st} queue, the next k_2 aircraft from the permutation assigned to the second queue and so on. For example, if n_T is 6, then a random permutation could be 2 5 1 4 6 3. If $l = 3$, then aircraft 2, 5 were assigned to the first queue, aircraft 1,4 were assigned to the second queue and aircraft 6,3 were assigned to the third queue.
- All the aircraft in each queue were ordered according to their increasing release times. For example, if aircraft 2,5,1 assigned to the same queue have their release times as 10,5,7 seconds respectively, then their ordering in the queue is $\{5, 1, 2\}$.

The algorithms were implemented on a Pentium 4, 3.00 GHz, 512 MB RAM. The computational results for three queues are presented in Fig. 3. The results show that a forty aircraft DSP can be solved to optimality, on an average, in less than one tenth of a second using the generalized dynamic programming approach (Fig. 3(a)). Also, the total aircraft delays corresponding to the optimal solutions are approximately twelve minutes lesser, on an average (Fig. 3(b)), than the delays corresponding to the FCFS sequences. Similar results are also presented for four queues in Fig. 4. These results show that the approach presented in this paper can produce optimal solutions relatively fast and can be used in a real-time simulation system.

VII. Possible Extensions to Variants of the DSP

***DSP* with time windows:**

In the DSP addressed in this paper, one of the constraints was that each aircraft can be scheduled only after its release time. In the DSP with time windows, a set of disjoint time intervals is given for each aircraft. Each aircraft can be scheduled only in any of the given set of intervals. This problem arises in the scenarios where a departure runway could be used by arrival aircraft for runway crossings. In this scenario, if specific time slots are assigned for the arrivals to use the runway, then each departing aircraft can be only assigned in a set of disjoint time intervals. The effect of these additional time constraints to the DSP is that the departure time defined in equation (7) may become infeasible for the new problem. To modify the approach to address a DSP with time window constraints, let $T_i \subseteq R^+$ denote the set of time intervals given for aircraft i. Note that T_i may be non-convex also. Now the set of all the non-dominated solutions can be calculated using exactly the same equations in (5) except that the definition of $LAST_s(m, k_1, \ldots, k_l)$ corresponding to a sequence s in equation (7) needs to be modified to the following one:

$$LAST_s(m, k_1, \ldots, k_l) = \begin{cases} \min\{t : t \in T_{a^m_{k_m}}\}, & \text{if } k_1 = k_2 = \ldots = k_l = 0, \\ \\ \min\{t : t \in T_{a^m_{k_m}}, t \geq LAST_r(n, k'_1, \ldots, k'_l) + sep(a^n_{k'_n}, a^m_{k_m})\}, & \text{otherwise.} \end{cases} \quad (9)$$

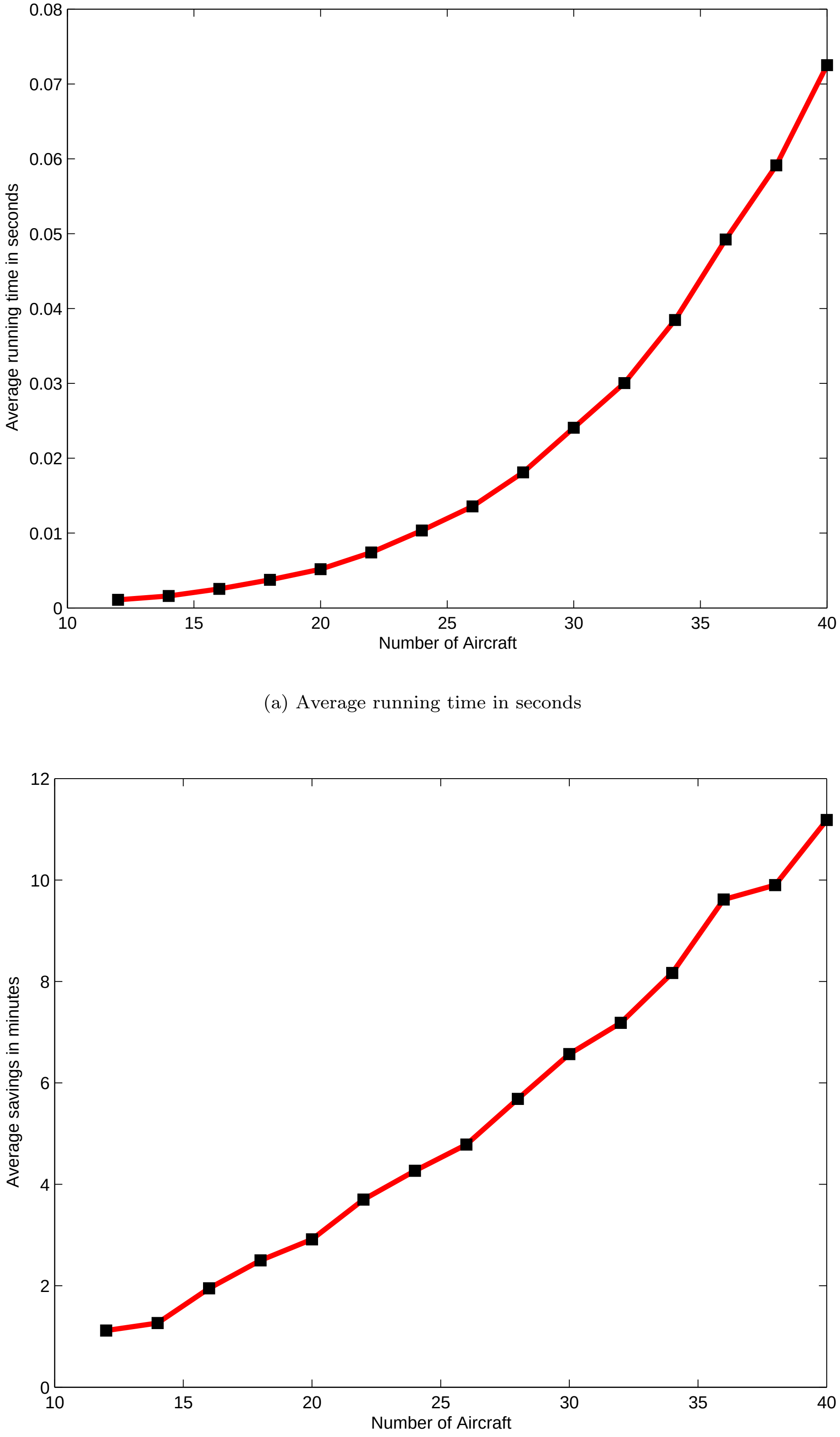
***DSP* with Constraint Position Shifting (CPS)**

As discussed in Section III, CPS restricts the position of the aircraft such that no aircraft can be shifted more than a given number of positions away from its corresponding position in the FCFS sequence. Let the position of aircraft i in the FCFS sequence be denoted by $FCFS_i$. Also, let POS^s_i be the position of aircraft i in a departure sequence s. If the maximum number of shifts is denoted by MNS, a DSP with CPS constraints requires that $|POS^s_i - FCFS_i| \leq MNS$ for s to be a feasible departure sequence. To use the generalized dynamic programming to solve the DSP with the CPS constraints, the equations given in (6) and (7) needs to be updated as follows:

$$\mathcal{F}(m, k_1, \ldots, k_l) = \begin{cases} \emptyset, & \text{if } |POS^s_y - FCFS_y| > MNS, y = a^m_{k_m}, \\ \\ \emptyset, & \text{if } |POS^s_y - FCFS_y| \leq MNS, y = a^m_{k_m} \text{ and } k_1 = \ldots = k_l = 0, \\ \\ \{s : s = (r, a^m_{k_m}), r \in \mathcal{F}_{nd}(n, k'_1, \ldots, k'_l), n \in Q\}, & \text{otherwise,} \end{cases} \quad (10)$$

where Q, k_i are defined in equations (2,3), and

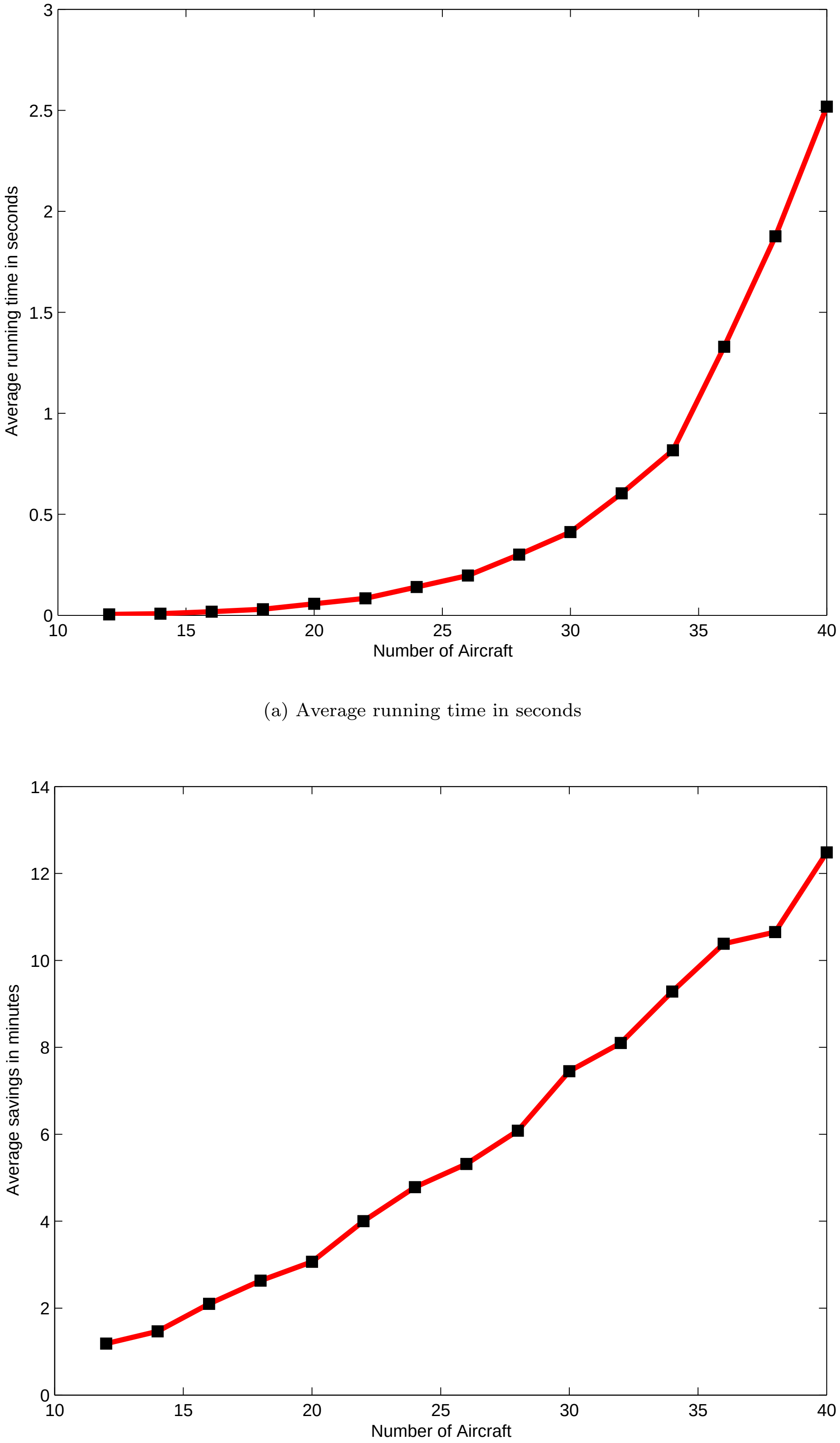
$$LAST_s(m, k_1, \ldots, k_l) = \begin{cases} \infty, & \text{if } |POS^s_y - FCFS_y| > MNS, y = a^m_{k_m}, \\ \\ \alpha(a^m_{k_m}), & \text{if } |POS^s_y - FCFS_y| \leq MNS, y = a^m_{k_m}, \text{ and } k_1 = \ldots = k_l = 0, \\ \\ \max(\alpha(a^m_{k_m}), LAST_r(n, k'_1, \ldots, k'_l) + sep(a^n_{k'_n}, a^m_{k_m})), & \text{otherwise,} \end{cases}$$



(a) Average running time in seconds

(b) Average savings using the optimal solution versus a FCFS solution in minutes

Figure 3. Results for the *DSP* with three queues.



(a) Average running time in seconds

(b) Average savings using the optimal solution versus a FCFS solution in minutes

Figure 4. Results for the *DSP* with four queues.

(11)

$$DELAY_s(m, k_1, \ldots, k_l) \quad = \quad DELAY_r(n, k'_1, \ldots, k'_l) + LAST_s(m, k_1, \ldots, k_l) - \alpha(a^m_{k_m}).$$

(12)

VIII. Conclusions

A generalized dynamic programming approach has been used to solve a departure scheduling problem that arises at airports. This approach finds optimal solutions to the departure scheduling problem with forty aircraft in less than one tenth of a second on an average. Also, the total aircraft delays corresponding to the optimal solutions are reduced by approximately twelve minutes on average compared to a first come, first served solution. Computational results seem to indicate that the approach presented in this paper is fast for a real-time implementation and can be used to reduce the aircraft delays at an airport. Future work can address generalizations of the departure scheduling problem discussed in this paper including problems with metering constraints for departure fixes, uncertainty etc.

References

[1] Carraway, R.L., and Morin, T.L., “Theory and applications of generalized dynamic programming: an overview”, International Journal of Computers and Mathematics with Applications, 16, 779-788, 1988.

[2] Atkins, S., Jung, Y., Stell, L., Brinton, C., and Rogowski, S., “Surface Management System Field Trial Results”, AIAA 4th Aircraft Technology, Integration, Integration, and Operations (ATIO) Forum, Chicago, Illinois, September 20-22, 2004.

[3] R.G. Dear and Y.S. Sherif, “The dynamic scheduling of aircraft in high density terminal areas, Microelect Reliab.”, 29 (1989), 743749.

[4] R.G. Dear and Y.S. Sherif, ”An algorithm for computer assisted sequencing and scheduling of terminal area operations“, Trans Res Part APolicy Pract 25 (1991), 129139.

[5] C.S. Venkatakrishnan, A. Barnett, and A.R. Odoni, “Landings at logan airport: Describing and increasing airport capacity, Trans Sci 27 (1993), 211227.

[6] J. Abela, D. Abramson, M. Krishnamoorthy, A. De Silva, and G. Mills, Computing optimal schedules for landing aircraft, Proc 12th National ASOR Conf, Adelaide, Australia, 1993, pp. 7190.

[7] J.A.D. Atkin, E.K. Burke, J.S. Greenwood, D. Reeson, “A metaheuristic approach to aircraft departure scheduling at London Heathrow Airport”, in Computer Aided Systems in Public Transport, Lecture Notes in Economics and Mathematical Systems, volume 600, (M.Hickman, P. Mirchandani, and S.Voss , Eds.) Springer 2008.

[8] J.A.D. Atkin, E.K. Burke, J.S. Greenwood, D. Reeson, “Hybrid Metaheuristics to Aid Runway Scheduling at London Heathrow Airport”, in Transportation Science, Volume 41, Number 1, pages 90-106, 2007.

[9] J.A.D. Atkin, E.K. Burke, J.S. Greenwood, D. Reeson, “An examination of take-off scheduling constraints at London Heathrow Airport”, in electronic proceedings of the 10th International Conference on Computer-Aided Scheduling of Public Transport , (CASPT2006), Leeds, UK, June 21-23, 2006.

[10] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha, and D. Abramson, “Scheduling aircraft landingsthe static case”, Trans Science.

[11] A.T. Ernst, M. Krishnamoorthy, and R.H. Storer, “Heuristic and exact algorithms for scheduling aircraft landings”. Networks, 34:229-241, 1999.

[12] L. Bianco, P. Dell’Olmo and S. Giordani, “Minimizing total completion time subject to release dates and sequence-dependent processing times”, Annals of Operations Research 86(1999), 393-415.

[13] L. Bianco, S. Ricciardelli, G. Rinaldi, and A. Sassano, “Scheduling tasks with sequence dependent processing times”, Naval Res Log 35 (1988), 177184.

[14] Ovacik, I.M. and Uzsoy, R., “Rolling Horizon Algorithms for a Single-Machine Dynamic Scheduling Problem with Sequence-Dependent Setup Times, International Journal of Production Research, Vol.32, pp.1243-1263 (1994).

[15] H.N. Psaraftis, “A dynamic programming approach for sequencing groups of identical jobs”, Oper Res 28 (1980), 13471359.

[16] Reha Uzsoy, Chung-Yee Lee, Louis A. Martin-Vega, “Scheduling semiconductor test operations: Minimizing maximum lateness and number of tardy jobs on a single machine”, Naval Research Logistics (NRL) Volume 39, Issue 3 , Pages 369 - 388.

[17] Torsten Fahle and Rainer Feldmann and Silvia Götz and Sven Grothklags and Burkhard Monien, “The Aircraft Sequencing Problem, Computer Science in Perspective: Essays Dedicated to Thomas Ottmann”, 2003, 152–166, Springer-Verlag New York, Inc., New York, NY, USA.

[18] H. Balakrishnan and B. Chandran, “Scheduling Aircraft Landings under Constrained Position Shifting,” AIAA Guidance, Navigation and Control Conference, Keystone, CO, August 2006.

[19] H. Lee and H. Balakrishnan, “Fuel Cost, Delay and Throughput Tradeoffs in Runway Scheduling,” Proceedings of the American Control Conference, June 2008.

[20] Min Wen, Jesper Larsen, Jens Clausen, “An exact algorithm for Aircraft Landing Problem”, Technical Report, Informatics and Mathematical Modelling, Technical University of Denmark”, DTU, September 2005.