Robust airline schedule design in a dynamic scheduling environment

Citation: Jiang, Hai and Barnhart, Cynthia"Robust Airline Schedule Design in a Dynamic Scheduling Environment." Computers & Operations Research 40, no. 3 (March 2013): 831–840. © 2011 Elsevier Ltd.

As Published: http://dx.doi.org/10.1016/j.cor.2011.06.018

Publisher: Elsevier

Persistent URL: http://hdl.handle.net/1721.1/108098

Version: Original manuscript: author's manuscript prior to formal peer review

Robust Airline Schedule Design In A Dynamic Scheduling Environment

Hai Jiang[a,*], Cynthia Barnhart[b]

[a]Department of Industrial Engineering, Tsinghua University, Beijing 100084, China
[b]Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Abstract

In the past decade, major airlines in the U.S. have moved from banked hub-and-spoke operations to de-banked hub-and-spoke operations to lower operating costs. In [1], it is shown that dynamic airline scheduling, an approach that makes minor adjustments to flight schedules in the booking period by re-fleeting and re-timing flight legs, can significantly improve utilization of capacity and hence increase profit. In this paper, we develop robust schedule design models and algorithms to generate schedules that facilitate the application of dynamic scheduling in de-banked hub-and-spoke operations. Such schedule design approaches are robust in the sense that the schedules produced can more easily be manipulated in response to demand variability when embedded in a dynamic scheduling environment. In our robust schedule design model, we maximize the number of *potentially connecting itineraries* weighted by their respective revenues. We provide two equivalent formulations of the robust schedule design model and develop a decomposition-based solution approach involving a variable reduction technique and a variant of column generation. We demonstrate, through experiments using data from a major U.S. airline, that the schedule generated can improve profitability when dynamic scheduling is applied. It is also observed that the greater the demand variability, the more profit our robust schedules achieve when compared to existing ones.

Keywords: airline schedule design, robust planning, dynamic scheduling, re-fleeting, re-timing

1. Introduction

Since 2001, the airline industry in the U.S. has been negatively impacted by terrorist attacks, soaring fuel cost, stiff competition, and a weak economy. In order to reduce operating costs, major airlines have decided to de-bank their banked schedules in their hub-and-spoke networks. A *hub-and-spoke network* is one in which, one or more, typically large, stations (or airports) are designated as hub stations and flights are scheduled between hub stations and the other stations, namely, spoke stations. A *banked schedule* refers to the practice where a set of aircraft arrive at the hub station from spoke stations in a relatively short period of time, park at the gates to allow passengers to deplane and connect to the next flights in their itineraries, and finally depart the hub station in a relatively short period of time. The set of inbound flights and outbound flights operated by these aircraft form a *bank* at the hub station. Airlines often schedule a series of banks at the hub station to capture time-of-day demand. Because flight arrivals and departures occur in a relatively short period to allow shorter passenger connection times, banked schedules generate negative, sometimes serious, economic impacts including demand peaks for various resources and increased schedule vulnerability in bad weather. In contrast, in de-banked operations, flight arrivals and departures at the hub are smoothed, eliminating the demand peaks for resources and reducing operating costs dramatically. American Airlines de-banked operations at its Chicago hub in April 2002, its Dallas/Fort Worth hub in November, 2002, and its Miami hub in May 2004 ([2, 3]). Continental Airlines de-banked its Newark hub ([3]); United Airlines de-banked its Chicago hub in 2004, its Los Angeles hub in 2005 ([4]); and Delta Airlines de-banked its Atlanta hub in January, 2005. This de-banking trend is not restricted to U.S. airlines. In 2004, Lufthansa Airlines de-banked Frankfurt, its biggest hub, as part of the effort to cut costs by EUR 300 million in two years ([5]).

[1] has shown that in a de-banked hub-and spoke network, dynamic scheduling, an approach that alters the flight schedule slightly according to observed actual demand and updated forecasts for future demand in the *booking period*, that is, the time from schedule publication until flight departure, can dramatically improve airline profitability. To date, two types of dynamic scheduling mechanisms (or schedule adjustment mechanisms) are studied in the literature: one is flight leg re-fleeting and the other is flight leg re-timing. Flight leg re-fleeting changes the fleet type assigned to a flight leg with higher than planned demand to a larger aircraft type, and the fleet type assigned to a flight leg with lower than planned demand to a smaller aircraft type while still maintaining aircraft flow balance. Representative literature describing flight leg re-fleeting can be found in [6], [7], and [8]. In flight leg re-timing, flight departure and arrival times of a flight leg are altered to create new connecting itineraries through the hub to serve markets with higher than expected demands. Later, [9, 10] present further experiments with dynamic scheduling under different scenarios, using data from a major European airline.

*Corresponding author
Email addresses: haijiang@tsinghua.edu.cn (Hai Jiang), cbarnhar@mit.edu (Cynthia Barnhart)

The success of dynamic scheduling not only relies on improved demand forecast quality, but also depends on the amount of re-optimization flexibility in the original schedule. One certainly can design a schedule where there is little or no room for dynamic scheduling. Hence, an important and challenging research question is how to design the flight schedule to maximize future dynamic scheduling impacts. In this paper, we address this question by designing *robust* schedule design models to produce schedules with improved capability to respond to demand uncertainty through dynamic scheduling. The result is that the schedules produced can better utilize the capacity and hence, capture more revenue.

Research aimed at improving schedule robustness in light of demand variability in a dynamic environment is rather limited. Focusing on models and solution approaches for the dynamic re-fleeting problem, [8] proposes to study the interaction between an original schedule and subsequent dynamic re-fleeting decisions. To our knowledge, the results of this study have not been reported to date. In a loosely related paper, [11] argues that suitably distributed aircraft capacity is critical to the successful implementation of dynamic re-fleeting procedures, and propose a scenario aggregation-based approach to determine an optimal fleet composition that facilitates dynamic re-fleeting.

Expanding on the observations of [6] that flight re-fleeting opportunities can be abundant in hub-and-spoke networks, we identify metrics to measure opportunities for flight re-timing. We focus on the magnitude of "infeasibility" for *infeasible connecting itineraries*, defined as connecting itineraries whose connection times are either less than the minimum required or more than the maximum allowed. When flights are re-timed in a dynamic scheduling environment to alter connecting times for itineraries, an infeasible connecting itinerary that violates the minimum (or maximum) connection time requirement by 5 minutes is easier to transform into a *feasible* itinerary than one that violates the minimum (or maximum) connection time requirement by 30 minutes.

In our schedule design model, we modify an existing de-banked schedule and generate a flight schedule in which the number of feasible and slightly infeasible connecting itineraries weighted by their respective revenues is maximized, while meeting two major conditions: 1) there is no banking (or peaking) in aircraft departures and arrivals at the hub, and 2) key connecting itineraries through hubs are not disrupted in the modified de-banked schedule. We present two alternate formulations of the robust model, each with different computational properties. To solve our models, we develop a decomposition-based approach involving a variable reduction technique and a variant of column generation. The schedule generated by the robust model is then simulated in a dynamic scheduling environment to evaluate its performance relative to the existing de-banked schedule.

The contributions of this paper include the following:

- We develop a novel approach to build flexibility into the original schedule to facilitate the application of dynamic scheduling;
- We present a mathematical model and two equivalent re-formulations to achieve robust schedules. We explore their computational properties and devise new solution algorithms; and
- We demonstrate with data from a major U.S. airline that our approach is capable of generating schedules that when combined with dynamic scheduling outperform the existing schedules.

We organize the remainder of this paper as follows. In Section 2, we present two alternative formulations of our robust schedule design model. We detail our solution algorithm in Section 3. In Section 4, we report the computational and dynamic scheduling results for our case study. Finally, In Section 5, we conclude our discussion and outline possible future research directions.

2. Robust Schedule Design

In order to increase the options and impacts of dynamic scheduling for a de-banked schedule, we consider the two dynamic airline scheduling elements, namely, flight leg re-fleeting and flight leg re-timing. Because a de-banked schedule maintains the hub-and-spoke network structure, numerous aircraft swapping opportunities at the hub are possible ([6]). Given this, we focus our efforts on creating flight leg re-timing opportunities that potentially can be executed during the dynamic scheduling stage.

In a hub-and-spoke network, an inbound flight leg f arriving at the hub at time t can connect to any outbound flight leg departing between $t + MinCT$ and $t + MaxCT$ to form a *feasible connecting itinerary*, where $MinCT$ represents the minimum passenger connection time and $MaxCT$ denotes the maximum passenger connection time at the hub. Any outbound flight departing earlier than $t + MinCT$ or later than $t + MaxCT$ forms an *infeasible connecting itinerary* with inbound flight leg f and contributes nothing toward revenue. In fact, virtually all schedule planning models in the literature, for example, [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24], have the objective to determine the optimal set of nonstop and feasible connecting itineraries to include in the schedule. For infeasible connecting itineraries, these models are indifferent to the extent of the *infeasibility*, that is, the amount of time these infeasible connecting itineraries violate the connection time requirements. We define *slightly infeasible connecting itineraries* as those connecting itineraries whose connection times violate the minimum or maximum connection time requirements by a small margin, such as 15 minutes or less. Slightly infeasible connecting itineraries have the property that they can be transformed into feasible connecting itineraries when flights are retimed in the dynamic scheduling process. Motivated by this observation, our robust schedule design model works to maximize the number of feasible and slightly infeasible connecting itineraries weighted by their respective revenues in the modified de-banked schedule. In doing so, we enhance the potential for dynamic scheduling (that is, flight leg re-timing) to increase capacity in markets experiencing greater than expected demand.

Note that maximizing the number of feasible and slightly infeasible connecting itineraries weighted by their respective revenues is equivalent to maximizing the total revenue of feasible and slightly infeasible connecting itineraries.

While achieving the goal mentioned above, our robust schedule design model has to satisfy many other constraints, including flight cover constraints, aircraft balance constraints, aircraft count constraints, de-banking constraints, and key connecting itinerary constraints. The latter two are uncommon and are explained in the following text. The de-banking constraints limit the number of departure and arrival activities at the hub to make sure that the new schedule is still de-banked. The key connecting itinerary constraints ensure that key connecting itineraries identified in the existing de-banked schedule are still feasible in the new schedule, that is, the connection times of those itineraries stay between the minimum connection time and the maximum connection time required at the hub. In our research, we classify connecting itineraries in the original de-banked schedule having at least five average daily passengers as key connecting itineraries. The idea is to ensure that our schedule design model does not disrupt important connecting itineraries while maximizing the total revenue of feasible and slightly infeasible connecting itineraries.

2.1. Terminology and Network Representations

To facilitate our discussion, we define the following terms. A *flight leg* is a nonstop trip of an aircraft from an origin airport to a destination airport (one take-off and one landing). An *itinerary* consists of a specific sequence of scheduled flight legs in which the first leg originates from the origin airport at a particular time and the final leg terminates at the final destination airport at a later time. The number of intermediate cities traversed is called the *number of stops* in this itinerary. A *nonstop itinerary* consists of only one flight leg, which originates from the origin and terminates at the destination. A *connecting itinerary* is an itinerary that has one or more stops. Although there do exist connecting itineraries with more than one intermediate stop, it is very rare. In the scope of our research, we assume all connecting itineraries have exactly one stop at the hub.

The aircraft flow network is used to model the flow of aircraft over a flight schedule, with a different flow network created for each fleet type. Each node in fleet π's network corresponds either to the departure time of a flight leg f, or its arrival time plus the minimum amount of time needed to turn a type π aircraft at the arrival station of leg f. Each arc in fleet π's network is classified as either a *flight arc* or a *ground arc*. Flight arcs represent scheduled flight legs, while *ground arcs* represent an aircraft's ability to remain on the ground at the same place over time. A *wrap-around arc* is a ground arc that connects the first and last node at an airport station. The *count line* is an arbitrarily chosen point of time that is used to count the number of aircraft needed to operate a given flight schedule.

The passenger flow network is used to model the flow of passengers over a flight schedule. Each node in the passenger flow network corresponds to either the departure time, or the arrival time of a flight leg. Each arc is classified as either a *flight arc* or a *connection arc*. Flight arcs represent scheduled flight legs, but *connection arcs* represent a passenger's ability to connect between two flight legs. In our robust schedule design model, connection arcs are also created for slightly infeasible connecting itineraries.

2.2. Model Statement

Input to the robust schedule design model includes the set of flight legs in an already de-banked schedule, each of which must be assigned to exactly one aircraft type. The departure time of the flight leg in the new schedule can differ from that of the original schedule, and these sets of decisions are modeled by creating a copy of each flight leg for every allowable departure time, and a set of constraints ensuring that exactly one copy is assigned one aircraft type.

We begin by enumerating all *potentially connecting itineraries*, that is, all itineraries whose connection times are *feasible* if allowable flight leg re-timings can be applied. Note that this set includes the set of connecting itineraries that are feasible without any re-timing as well as those slightly infeasible itineraries. The set of potentially connecting itineraries can thus be defined as all itineraries with connection times between T_1 and T_2, where $T_1 < MinCT$ and $T_2 > MaxCT$, and the precise values of T_1 and T_2 are a function of the maximum allowable amount of time that flight legs can be shifted in the dynamic scheduling process. For example, if $MinCT = 25$ minutes, $MaxCT = 180$ minutes, and the maximum amount of re-timing is 15 minutes, we then have $T_1 = 25 - 15 = 10$ minutes and $T_2 = 180 + 15 = 195$ minutes. Let PC be the set of all potentially connecting itineraries. We assign a binary *connection variable* h_p with weight w_p to each potentially connecting itinerary $p \in PC$, where w_p is the revenue associated with the connecting itinerary corresponding to p, assuming that p is a valid connecting itinerary. Note that for potentially connecting itineraries created between copies of the same flight leg pair, their w_p values are the same. Mathematically, let $w_{(l_1,l_2)}$ be the revenue associated with connecting itinerary formed by l_1 and l_2, then for p_1, p_2 formed by copies of l_1 and l_2, $w_{p_1} = w_{p_2} = w_{(l_1,l_2)}$. The exactly value of $w_{l_1,l_2)}$ is estimated using historical demand data. Binary variable h_p takes value one if both flight leg copies forming potentially connecting itinerary p are selected in the solution; otherwise, $h_p = 0$. The objective, then, is to maximize the number of potentially connecting itineraries weighted by their respective revenues, that is, to maximize the total revenue associated with these variables representing potentially connecting itineraries:

$$\text{maximize} \sum_{p \in PC} w_p h_p.$$

Before detailing our robust schedule design model, we introduce the following notations:

Additional Data and Parameters

L : set of flight legs in the flight schedule indexed by l.

$C(l)$: set of flight copies for flight leg $l \in L$.

a : the number of flight copies created before and after the original flight leg.

l^k: copy $k \in C(l)$ of flight leg $l \in L$.
Π : set of fleet types indexed by π.
G^π : set of ground arcs in fleet $\pi \in \Pi$'s network.
N^π : set of nodes in flight network of fleet type $\pi \in \Pi$.
n^π : number of aircraft available of fleet type $\pi \in \Pi$.
T : set of time intervals at the hub, indexed by t.
MAX^{at}: maximum number of aircraft arrivals at the hub in interval $t \in T$.
MAX^{dt}: maximum number of aircraft departures from the hub in interval $t \in T$.

$$\overline{\alpha}^i_{lk\pi}=\begin{cases} 1, & \text{if } l^k \text{ in fleet } \pi\text{'s network originates from node } i \in N^\pi, \forall k \in C(l), l \in L, \pi \in \Pi; \\ -1, & \text{if } l^k \text{ in fleet } \pi\text{'s network terminates at node } i \in N^\pi, \forall k \in C(l), l \in L, \pi \in \Pi; \\ 0, & \text{otherwise.} \end{cases}$$

$$\widehat{\alpha}^i_{g\pi}=\begin{cases} 1, & \text{if ground arc } g \in G^\pi \text{ originates from node } i \in N^\pi, \forall \pi \in \Pi; \\ -1, & \text{if ground arc } g \in G^\pi \text{ terminates at node } i \in N^\pi, \forall \pi \in \Pi; \\ 0, & \text{otherwise.} \end{cases}$$

$$\overline{\beta}_{lk\pi}=\begin{cases} 1, & \text{if } l^k \text{ in fleet } \pi\text{'s network crosses the count line, } \forall k \in C(l), l \in L, \pi \in \Pi; \\ 0, & \text{otherwise.} \end{cases}$$

$$\widehat{\beta}_{g\pi}=\begin{cases} 1, & \text{if ground arc } g \in G^\pi \text{ crosses the count line, } \forall \pi \in \Pi; \\ 0, & \text{otherwise.} \end{cases}$$

$$\gamma^{at}_{lk}=\begin{cases} 1, & \text{if } l^k \text{ in the passenger flow network arrives at the hub during interval } t \in T, \forall k \in C(l), l \in L; \\ 0, & \text{otherwise.} \end{cases}$$

$$\gamma^{dt}_{lk}=\begin{cases} 1, & \text{if } l^k \text{ in the passenger flow network departs from the hub during interval } t \in T, \forall k \in C(l), l \in L; \\ 0, & \text{otherwise.} \end{cases}$$

(l_1, l_2) : ordered pair of inbound flight leg $l_1 \in L$ and outbound flight leg $l_2 \in L$ at the hub.
C : set of all flight leg pairs (l_1, l_2).
$(l_1^{k_1}, l_2^{k_2})$: a potentially connecting itinerary formed by $l_1^{k_1}$ and $l_2^{k_2}$.
$C(l_1, l_2)$: the set of potentially connecting itineraries formed between copies of flight leg l_1 and copies of flight leg l_2.
PC:the set of all potentially connecting itineraries, indexed by p.

$$\zeta_p^{l^k} = \begin{cases} 1, & \text{if } p \in PC \text{ traverses } l^k; \\ 0, & \text{otherwise.} \end{cases}$$

Q: the set of key connecting itineraries identified in the original de-banked schedule that must be feasible in the schedule generated by our robust schedule design model. It contains a collection of flight leg pairs (l_1, l_2) through the hub. We need to ensure that in the new schedule, l_1 can connect to l_2 for all $(l_1, l_2) \in Q$.
$CT(l_1^{k_1}, l_2^{k_2})$: the connection time between $l_1^{k_1}$ and $l_2^{k_2}$, that is, the connection time between copy k_1 of flight leg l_1 and copy k_2 of flight leg l_2.

Additional Decision Variables

$$f_{lk\pi}=\begin{cases} 1, & \text{fleet } \pi \in \Pi \text{ is used to fly flight copy } l^k, \forall k \in C(l), l \in L; \\ 0, & \text{otherwise.} \end{cases}$$

$y_{g\pi}$: number of aircraft of fleet type π traversing ground arc $g \in G^\pi, \forall \pi \in \Pi$.

2.3. Formulation 1

Now we develop the constraints for the robust schedule design model. In Figure 1 we show an inbound flight leg l_1 and an outbound flight leg l_2, each with three copies in the passenger flow network. Three connection arc variables are illustrated, that is, (l_1^a, l_2^d), (l_1^a, l_2^e), (l_1^a, l_2^f) and are indexed as h_1, h_2, and h_3. For each flight leg copy l^k, if the term $\sum_{\pi\in\Pi} f_{lk\pi}$ equals one, this flight leg copy is assigned an aircraft type, or in other words, selected, in the solution; otherwise, it is not.

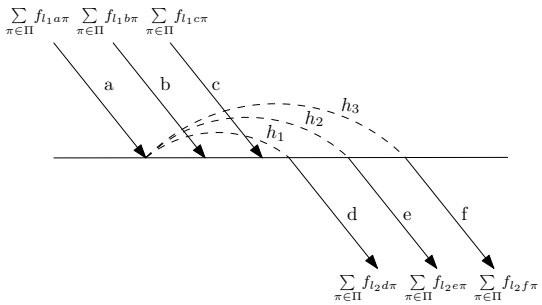

Figure 1: Illustration of connection variables

Take the example shown in Figure 1, Formulation 1 generates three constraints for the three connection variables involving l_1^a, specifically,

$$h_1 \le \sum_{\pi\in\Pi} f_{l_1a\pi},$$

$$h_2 \le \sum_{\pi\in\Pi} f_{l_1a\pi},$$

$$h_3 \le \sum_{\pi\in\Pi} f_{l_1a\pi}.$$

Recognizing the fact that at most one of h_1, h_2, and h_3 can be non-zero in any feasible integer solution, the following constraint must hold:

$$h_1 + h_2 + h_3 \le \sum_{\pi\in\Pi} f_{l_1a\pi}.$$

Hence, for all $p \in C(l_1, l_2)$ ($(l_1, l_2) \in C$), the following constraints must be satisfied:

$$\sum_{p\in C(l_1,l_2)} \zeta_p^{l_1^k} h_p \le \sum_{\pi\in\Pi} f_{l_1k\pi}, \forall k \in C(l_1) \tag{1}$$

and

$$\sum_{p\in C(l_1,l_2)} \zeta_p^{l_2^k} h_p \le \sum_{\pi\in\Pi} f_{l_2k\pi}, \forall k \in C(l_2) \tag{2}$$

Constraints (1) require that for flight leg copy l_1^k, if any of the h_p variables that traverses l_1^k takes value 1, one aircraft is assigned to fly l_1^k. Constraints (2) ensure that for flight leg copy l_2^k, if any of the h_p variables that traverses l_2^k takes value 1, one aircraft is assigned to fly l_2^k. The remaining constraints for the robust schedule design model are as follows:

$$\sum_{k\in C(l)} \sum_{\pi\in\Pi} f_{lk\pi} = 1, \forall l \in L \tag{3}$$

$$\sum_{l\in L}\sum_{k\in C(l)}\overline{\alpha}^{i}_{lk\pi}f_{lk\pi}+\sum_{g\in G^{\pi}}\widehat{\alpha}^{i}_{g\pi}y_{g\pi}=0,$$

$$\forall i\in N^{\pi},\pi\in\Pi \quad (4)$$

$$\sum_{l\in L}\sum_{k\in C(l)}f_{lk\pi}\overline{\beta}_{lk\pi}+\sum_{g\in G^{\pi}}y_{g\pi}\widehat{\beta}_{g\pi}\le n^{\pi},\forall\pi\in\Pi \quad (5)$$

$$\sum_{l\in L}\sum_{k\in C(l)}\gamma^{at}_{lk}\sum_{\pi\in\Pi}f_{lk\pi}\le MAX^{at},\forall t\in T \quad (6)$$

$$\sum_{l\in L}\sum_{k\in C(l)}\gamma^{dt}_{lk}\sum_{\pi\in\Pi}f_{lk\pi}\le MAX^{dt},\forall t\in T \quad (7)$$

$$\sum_{\pi\in\Pi}f_{l_1k_1\pi}+\sum_{\pi\in\Pi}f_{l_2k_2\pi}\le 1,$$

$$\forall(l_1,l_2)\in Q, CT(l_1^{k_1},l_2^{k_2})\notin[MinCT,MaxCT] \quad (8)$$

$$f_{lk\pi}\in\{0,1\},\forall l\in L,k\in C(l),\pi\in\Pi \quad (9)$$

$$y_{g\pi}\ge 0,\forall g\in G,\pi\in\Pi \quad (10)$$

$$h_p\in\{0,1\},\forall p\in PC \quad (11)$$

Constraints (3) ensure that each flight leg is covered exactly once, while Constraints (4) enforce conservation of flow for each type of aircraft. Constraints (5) count and limit the number of aircraft of each fleet used to the number available. Constraints (6) and (7), the *de-banking* constraints, limit the number of flight departures and arrivals per unit time at the hub. Constraints (8) guarantee that key connecting itineraries identified in the original de-banked schedule remain feasible in the new schedule. For all key connecting itineraries, we forbid the selection of flight copy pairs that do not meet the connection time requirements. Constraints (9) through (11) specify the possible values of the decision variables. We refer to this formulation as Formulation 1.

2.4. Formulation 2

In this section, we present an equivalent formulation whose LP relaxation is weaker than that of Formulation 1, but which exhibits better computational performance using branch-and-bound in our solution approach.

The connection variable h_p can take value zero or one, thus we can think of sending one unit of flow on potentially connecting itinerary p and have a pseudo-capacity (M_{lk}) for each flight copy l^k. Take the example in Figure 2. Depicted are one inbound flight leg l_1, two outbound flight legs l_2 and l_3, each with three copies, and six connection variables h_1, h_2, $\cdots$, and h_6. The six connection variables all traverse l_1^a. If any of the six connection variables take value one in a feasible integer solution, $\sum_\pi f_{l_1a\pi}$ must take value one. Alternatively, if $\sum_\pi f_{l_1a\pi}$ takes value zero, none of the six connection variables can take value one. Such a relationship can be captured by the following constraint:

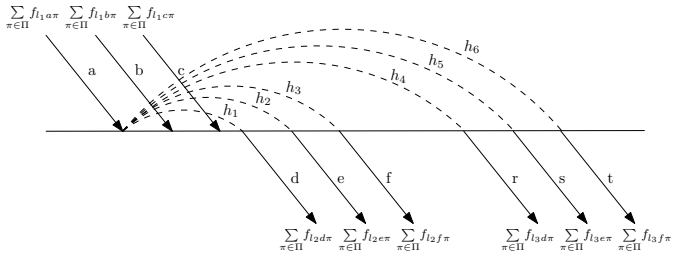

Figure 2: Illustration of Formulation 4

$$h_1+h_2+\cdots+h_6\le M_{l_1a}\sum_{\pi\in\Pi}f_{l_1a\pi}. \quad (12)$$

M_{l_1a} represents a sufficiently large number (the pseudo capacity of l_1^a), which guarantees that if $\sum_{\pi\in\Pi}f_{l_1a\pi}=1$, Constraint (12) is not binding. Following similar logic, we replace Constraints (1) and (2) in Formulation 1 with Constraints (13) to model the relationship between h_p variables and $f_{lk\pi}$ variables in Formulation 2. The left hand side sums over all potential connecting itineraries that traverse l^k. If any of the h_p takes value 1, we must assign an aircraft type to l^k:

$$\sum_{p\in PC}\zeta_p^{l^k}h_p\le M_{lk}\sum_{\pi\in\Pi}f_{lk\pi},\forall l\in L,k\in C(l). \quad (13)$$

3. Solution Approach

The solution approach to the robust schedule design model is outlined in Figure 5. For reasons stated in Section 4.1, Formulation 2 is used to find good integer solutions, while the LP relaxation of Formulation 1 is solved to obtain a bound (Z_L) on the optimal integer solution value and to gauge the optimality of the integer solutions obtained from Formulation 2.

To solve Formulation 2, a decomposition-based approach is taken in which a *Restricted Master Problem* (*RMP*) containing only a subset of all possible h_p variables is solved repeatedly until a near-optimal solution to the model is obtained. The rationale underlying the selection of the variables included in the *RMP* is as follows. We first classify flight leg pairs $(l_1,l_2)\in C$ as follows. Recall that we have defined $C(l_1,l_2)$ as the set of potentially connecting itineraries formed between the flight leg copies of l_1 and l_2. We classify all such flight leg pairs (l_1,l_2) according to the size of $C(l_1,l_2)$, that is, the number of potentially connecting itineraries formed between flight leg copies of l_1 and l_2. Let C_b be the set of flight leg pairs (l_1,l_2), where $|C(l_1,l_2)|=b$ ($b\in\mathbb{N}$), that is, $C_b=\{(l_1,l_2)\mid|C(l_1,l_2)|=b,(l_1,l_2)\in C\}$. Since a is the number of flight leg copies created before and after the original flight leg copy, respectively, we know that b must stay between 1 and $(2a+1)^2$.

For any flight leg pair $(l_1,l_2)\in C_{(2a+1)^2}$, regardless of which copies of flight leg l_1 and l_2 are selected in an integer solution, they form a valid connection with the same contribution $w_p=w_{(l_1,l_2)}$ toward the objective function value. The removal of all columns corresponding to h_p, $p\in P(C_{(2a+1)^2})$ thus have the effect of reducing the objective function value for any solution by a constant amount. The result is that an optimal solution to the *RMP* is also optimal to the original model containing all decision variables. This idea is further illustrated in Figure 3. There are three flight leg copies of l_1 and three flight leg copies of l_2. Altogether they create $(2\times1+1)^2=9$ potentially connecting itineraries and their coefficients in the objective function are

the same, that is, $w_{(l_1,l_2)}$. Hence, in any feasible integer solution, independent of which of the three flight leg copies of l_1 is selected, and which of the three flight leg copies of l_2 is selected, there is exactly one h_p variable that takes value 1 and the collective contribution of all h_p variables created between copies of l_1 and l_2 is $w_{(l_1,l_2)}$, which is a constant. The removal of all h_p variables created between copies of l_1 and l_2 thus will not affect the values of schedule decision variables, that is, $f_{lk\pi}$ variables, in our optimal solution.

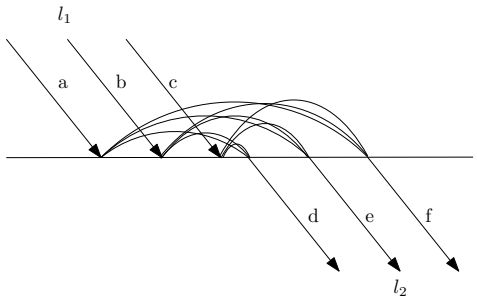

Figure 3: When $(l_1, l_2) \in C_{(2a+1)^2}$, that is, all pairs of flight leg copies between l_1 and l_2 form potentially connecting itineraries, we can remove all of these h_p variables without affecting the values of $f_{lk\pi}$ in the optimal solution. In this example, $a = 1$, that is, the number of flight leg copies created before and after the original flight leg is 1.

Extending this approach, we can remove all columns corresponding to h_p with $p \in P(C_{(2a+1)^2-1})$, however, we can no longer guarantee that an optimal solution to *RMP* is optimal for the original problem because the solution to the *RMP* might select, from all $(2a + 1)^2$ potential connections, the *only* pair of flight leg copies that does not create a potentially connecting itinerary. The risk of doing so is minimal but increases as the subscript b in C_b decreases. This is illustrated in Figure 4, where all pairs of flight leg copies between l_1 and l_2 form potentially connecting itineraries, except for the pair between l_1^a and l_2^f. If we drop all h_p variables corresponding to l_1 and l_2 in *RMP*, the optimal solution will be affected if l_1^a and l_2^f are selected in the optimal solution to the full problem of our robust schedule design model. We therefore proceed as follows: in the beginning, we exclude these h_p variables from the model and form our *RMP*. Once we find the optimal solution to *RMP*, however, we check whether potentially connecting itinerary $p \in C(l_1, l_2)$ is enabled or not in the solution. If it is, we set h_p to 1; otherwise, we set h_p to 0. Now, we check whether $\sum_{p \in C(l_1,l_2)} h_p = 1$. If the answer is yes, this optimal solution to *RMP* is also an optimal solution to the full problem. If not, for example, when l_1^a and l_2^f are selected in the optimal solution and the eight h_p variables all take value zero, we do not have $\sum_{p \in C(l_i,l_j)} h_p = 1$. Thus, we then add the h_p variables formed between copies of l_1 and l_2 to the *RMP* and resolve *RMP*.

To generalize the above idea, we let $\widetilde{C} = C_{(2a+1)^2} \bigcup C_{(2a+1)^2-1} \bigcup \cdots \bigcup C_b$ and $P(\widetilde{C})$ be the set of potentially connecting itineraries associated with $(l_1, l_2) \in \widetilde{C}$. We first eliminate all h_p variables with $p \in P(\widetilde{C})$ in Formulation 2 to create *RMP*. The branch-and-bound algorithm is used to find integer solutions to *RMP*. Once an integer solution is found, we denote the

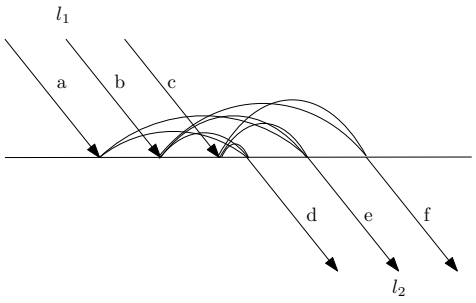

Figure 4: When $(l_1, l_2) \in C_{(2a+1)^2-1}$, that is, all but one pair of flight leg copies between l_1 and l_2 are potentially connecting itineraries, we can remove all of these h_p variables with some risk. In this example, flight copies l_1^a and l_2^f do not form a potentially connecting itinerary.

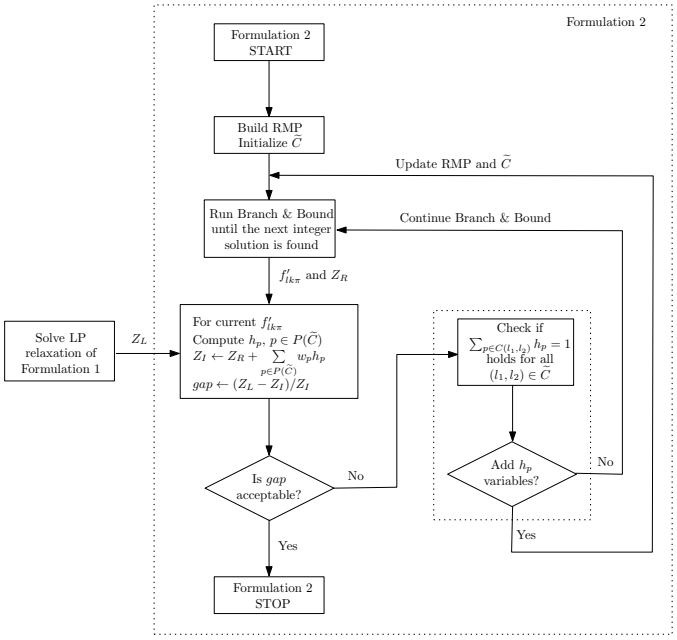

Figure 5: Solution algorithm for the robust schedule design model

objective function value of *RMP* as Z_R and let its solution for $f_{lk\pi}$ variables be denoted by $f'_{lk\pi}$. If connection $p \in P(\widetilde{C})$ is enabled in this current integer solution, we set h_p to 1; otherwise, we set h_p to 0, for each $p \in P(\widetilde{C})$. The objective function value of the current feasible solution for Formulation 2, denoted Z_I is then equal to $Z_R + \sum_{p \in P(\widetilde{C})} w_p h_p$. We compute the optimality gap as $gap = (Z_L - Z_I)/Z_I$. If the optimality gap is acceptable, that is, within our defined threshold value, the algorithm terminates; otherwise, we compute the value of $\sum_{p \in C(l_1,l_2)} h_p$, for each $(l_1, l_2) \in \widetilde{C}$. For each $(l_1, l_2) \in \widetilde{C}$ for which $h_{(l_1,l_2)} \neq 1$, all columns corresponding to h_p, $p \in C(l_1, l_2)$ are added to *RMP* and $\widetilde{C}$ is updated as follows: $\widetilde{C} \leftarrow \widetilde{C} \setminus (l_1, l_2)$. Next, *RMP* is resolved. If $h_{(l_1,l_2)} = 1$ holds true for all $(l_1, l_2) \in \widetilde{C}$, we continue the branch-and-bound algorithm to identify another feasible solution to Formulation 2.

This approach is different from conventional column generation in that the columns excluded from the *RMP* include those that appear in the basic feasible solution, that is, we exclude

columns whose corresponding decision variables are nonzero in the optimal solution. The consequence is that although the optimal solution to our RMP is the same as that to the original problem, the optimal objective function value of our *RMP* can be different. In conventional column generation, the columns excluded are those that do not appear in the basic feasible solution; hence, not only is the optimal solution to the restricted problem the same as that to the original problem, but also the optimal objective function value of the restricted problem is the same.

4. Case Study

In this section, using data obtained from a major U.S. airline, we compare the performance of the robust de-banked schedule to that of an existing de-banked schedule in a dynamic scheduling environment. The airline providing us data operates a hub-and-spoke network with approximately 1000 flight legs serving about 100 cities daily, and about 300 flight legs departing from and 300 flight legs arriving at the major hub each day. The number of departures and the number of arrivals are each limited to 5 per 10-minute interval, respectively.

In the robust schedule design model, seven copies of each flight leg l are created (that is, $a = 3$), placing one each at -30, -20, -10, 0, +10, +20, and +30 minutes offset from leg l's scheduled departure time in the original schedule.According to rules from the airline, $MinCT$ is set to 25 minutes and $MaxCT$ is set to 180 minutes.

In our experiments with the robust schedule design model, we consider as potentially connecting itineraries those that violate the minimum or maximum connection time requirement by 15 minutes. That is, $T_1 = MinCT - 15$ minutes and $T_2 = MaxCT + 15$ minutes. In dynamic airline scheduling, those slightly infeasible itineraries with connection times between T_1 and $MinCT$ can be made feasible, that is, their connection times can become slightly longer than $MinCT$, if flight legs are re-timed. Similarly, those slightly infeasible itineraries with connection times between $MaxCT$ and T_2 can be made feasible, that is, their connection times can become slightly shorter than $MaxCT$, if flight legs are re-timed.

4.1. Computational Results

The robust schedule design model is implemented in C using ILOG CPLEX 9.0. Computational experiments are conducted on a workstation equipped with one Intel Pentium 4 2.8 GHz processor and 1 GB RAM.

In Table 1, we report, for Formulations 1 and 2, the sizes of the models after CPLEX preprocessing [see 25, p. 322-324], and the optimal objective function values of the corresponding LP relaxations. Formulation 1 provides tighter bound than Formulation 2, and hence, we use it to generate bounds with which to measure the optimality gap of integer solutions.

Table 2 summarizes the computational performance of Formulations 1 and 2 during branch-and-bound. We can see that the number of rows, columns, and non-zeros of the *RMP* are significantly reduced when compared to those reported in Table 1. The fourth row reports the number of fractional variables at the root node. The fifth row reports the number of nodes searched in the branch-and-bound tree until the first integer solution is found. Formulation 1 fails to find integer solutions in 10 hours, the maximum allowable solution time. The sixth row reports the total number of nodes searched in 10 hours. Clearly, when exploring the branch-and-bound tree, Formulation 2 performs much better than Formulation 1. The next section of the table reports the total number of h_p variables in the full problem of Formulation 2 and the number of such variables included in the final *RMP*. About 15.9% of them are included in final *RMP*. When the algorithm terminates, the objective function value of Formulation 2 is 126,217 with an optimality gap of 2.32%.

	Formulation 1	Formulation 2
Num. of Rows	221,858	36,408
Num. of Cols.	514,834	514,921
Num. of NZ.	1,200,925	1,469,236
Z_L	129,142	142,739

Table 1: Comparison of LP relaxations for Formulations 1 and 2 for the full problem after CPLEX preprocessing

	Fm. 1	Fm. 2
Statistics of the Initial *RMP*		
Num. of rows	98,363	24,431
Num. of columns	193,741	193,803
Num. of non-zeros	436,733	504,118
Num. of fractionl var. at root node	3,225	3,572
Num. of node searched until first integer solution	Not found	760
Total num. of nodes searched in 10 hrs.	127	9,037
Total number of h_p variables		872,193
Number of h_p variables included in final *RMP*		139,051
Z_L from Formulation 1	129,142	
Z_I from Formulation 2		126,217
Optimality Gap $(Z_L - Z_I)/Z_I$		2.32%

Table 2: Branch-and-bound results for Formulations 1 and 2. Problem sizes are reported after CPLEX preprocessing

4.2. Dynamic Scheduling Results

For the existing de-banked schedule and the robust schedule generated by our robust schedule design model, we conduct the dynamic scheduling experiments detailed in [1] for operations spanning one week. The process is illustrated in Figure 6. The booking period is divided into two periods by the one-time schedule re-optimization point. We first flow unconstrained passenger demands in Period 1 through both the original schedule and the robust schedule. Then, using the schedule re-optimization model (which is part of the dynamic scheduling process), we re-optimize the original schedule and the robust schedule separately using the same updated demand forecast. Next, we flow unconstrained passenger demands in Period 2 through the modified schedules and obtain the final passenger profit at departure date for both schedules. Finally, we compare passenger profits for each schedule to gauge the benefit of our robust schedule design model.

According to results reported in [1, 10], we set the dynamic airline scheduling parameters as follows: a one-time re-optimization point is set 21-days prior to departure; the number

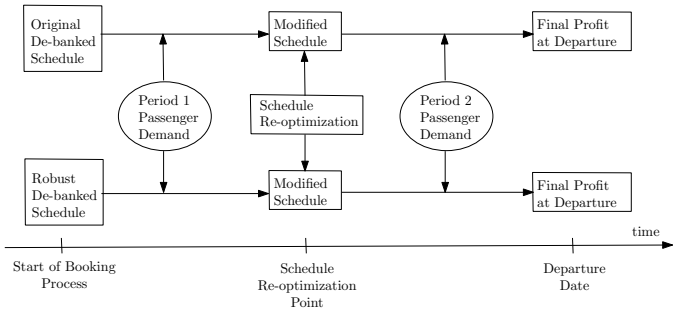

Figure 6: Illustration of the dynamic scheduling experiment

of re-timed flight legs is limited to 100 per day; and the number of re-fleeted flight legs is unconstrained. When we re-optimize the flight schedule in the dynamic scheduling process, we use historical demand averages as the forecasts for future demand. Because airlines typically utilize more sophisticated, and hence more accurate, forecasting engines, the schedule's performance under this scenario provide an estimate of a lower bound on the expected impacts.

The unconstrained demand in a given market for Period i ($i = 1, 2$) is sampled from a normal distribution with parameters estimated from historical data. In a given market, we model the unconstrained demand for market m as a normal distribution with mean D_m^i and variance σ_m^i, where D_m^i and σ_m^i are estimated from historical data. We then generate the simulated demand for market m in period i, D_m^{is} by drawing from the following normal distribution $N(D_m^i, \sigma_m^i)$. In Table 3, we show the daily profits for the original schedule and the robust schedule in a week's operation. For each individual day in this week, we generate 20 instances of demand and report the average profit for the original schedule (Column 2), the average profit for the robust schedule (Column 3), the absolute change in profitability (Column 4), and the percentage change in profitability (Column 5). In the last row, we average the numbers across all seven days. On average, the daily profit generated in the robust schedule is \$57,917 or 2.81% higher than that in the original de-banked schedule. These results show that our de-banking model successfully generates a more robust schedule that better handles demand uncertainty, that is, generates higher profit, in a dynamic scheduling environment.

Day	Original	Robust	Profit Incr.	Pct. Incr.
1	2,871,652	2,920,183	48,531	1.69%
2	2,183,397	2,260,034	76,637	3.51%
3	1,089,048	1,117,908	28,860	2.65%
4	1,475,938	1,523,315	47,378	3.21%
5	2,271,185	2,326,829	55,644	2.45%
6	2,837,982	2,935,609	97,627	3.44%
7	1,297,725	1,348,466	50,741	3.91%
Average	2,003,847	2,061,764	57,917	2.81%

Table 3: Comparisons of schedule profitability between the original schedule and the robust schedule for each individual day in a week's operation (in dollars).

Because the goal of our robust scheduling model is to create more potentially connecting itineraries to facilitate the handling of demand variability in a dynamic scheduling context, we further test the performance of our robust schedule when demand variability changes. Thus, instead of sampling from $N(D_m^i, \sigma_m^i)$, we sample from $N(D_m^i, \mu\sigma_m^i)$, where $\mu > 0$ controls the variability of the simulated demand. For the one week's operation, we first reduce demand variability by setting $\mu = 0.8$ and later increase demand variability by setting $\mu = 1.2$. Table 4 summaries the results when $\mu = 0.8$. We observe that the average daily profitability increased by \$28,067 or 1.44%. This indicates that although our robust schedule still outperforms the original schedule, the amount of improvement gets smaller because of less demand variability. In Table 5, we report the results when μ takes value 1.2. The robust schedule delivers much greater lift in profitability: the average daily profit increase is \$90,772 or 4.20%. This indicates that the greater the demand variability, the more our robust schedule outperforms the existing schedule.

Day	Original	Robust	Profit Incr.	Pct. Incr.
1	3,181,886	3,235,342	53,456	1.68%
2	2,161,779	2,198,313	36,534	1.69%
3	939,341	947,513	8,172	0.87%
4	1,408,421	1,430,814	22,394	1.59%
5	2,106,008	2,140,968	34,960	1.66%
6	2,603,654	2,628,909	25,255	0.97%
7	1,245,816	1,261,513	15,697	1.26%
Average	1,949,558	1,977,625	28,067	1.44%

Table 4: Comparisons of schedule profitability between the original schedule and the robust schedule for each individual day in a week's operation (in dollars). $\mu = 0.8$

Day	Original	Robust	Profit Incr.	Pct. Incr.
1	3,570,874	3,764,059	193,184	5.41%
2	2,199,913	2,303,748	103,836	4.72%
3	965,793	1,012,634	46,841	4.85%
4	1,556,112	1,594,703	38,592	2.48%
5	2,475,592	2,537,234	61,642	2.49%
6	2,425,415	2,541,107	115,692	4.77%
7	1,297,076	1,372,696	75,620	5.83%
Average	2,070,111	2,160,883	90,772	4.20%

Table 5: Comparisons of schedule profitability between the original schedule and the robust schedule for each individual day in a week's operation (in dollars). $\mu = 1.2$

5. Conclusion and Future Research Directions

In this paper, we develop a robust schedule design model to achieve maximal impact in a dynamic scheduling environment. To the best of our knowledge, it is the first approach of this type. We design a robust schedule design model to maximize the number of potentially connecting itineraries weighted by their associated revenues. Two equivalent formulations of this model are presented and studied. We show through experiments using data from a major U.S. airline, that the schedule generated by the robust schedule design model can achieve profit improvement. Results also show that the greater the demand variability, the greater the increase in profitability our robust schedule design approach achieves.

Potential future research directions include:

- Define additional metrics to measure schedule robustness. In this paper, we use the number of potentially connecting itineraries weighted by their respective revenues as the metric defining schedule robustness. An important research topic is to specify other metrics, especially those that integrate re-timing and re-fleeting impacts, and measure their effectiveness in creating robust schedules from which realized profits can be maximized.
- Build a feedback loop between robust planning models and dynamic scheduling models. The original schedule largely defines the set of feasible dynamic scheduling decisions; but dynamic scheduling decisions provide valuable information about the quality of the original schedule. If profits improve by frequently and consistently re-timing or re-fleeting particular flight legs, it might be possible to modify the original schedule to reduce the need to re-time and re-fleet those flight legs. A challenging research question is to assess the potential for improving the original schedule, using a feedback loop and simulations with our dynamic and robust scheduling approaches.

Acknowledgement

We are indebted to two anonymous referees for their invaluable comments that have allowed us to improve the quality of this manuscript.

References

[1] H. Jiang, C. Barnhart, Dynamic airline scheduling, Transportation Science 43 (3).
[2] P. Flint, No peaking, Air Transport World 39 (11) (2002) 22–27.
[3] J. Ott, 'de-peaking' american hubs provides network benefit, Aviation Week & Space Technology.
[4] UAL corporation reports fourth quarter and full year 2005 results.
URL http://www.united.com/press/detail/0,6862,53614-1,00.html
[5] J. Flottau, Lufthansa reorganizes flight operations, 'depeaks' frankfurt, Aviation Daily (2003) 3.
[6] M. E. Berge, C. A. Hopperstad, Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms, Operations Research 41 (1) (1993) 153–168, special Issue on Stochastic and Dynamic Models in Transportation.
[7] E. K. Bish, R. Suwandechochai, D. R. Bish, Strategies for managing the flexible capacity in the airline industry, Naval Research Logistics 51 (2004) 654–684.
[8] H. D. Sherali, E. K. Bish, X. Zhu, Polyhedral analysis and algorithms for a demand-driven refleeting model for aircraft assignment, Transportation Science 39 (3) (2005) 349.
[9] T. Hansen, V. Warburg, Dynamic airline scheduling - analyses, enhancements, and experiments, Master's thesis, Technical University of Denmark (2006).
[10] V. Warburg, T. G. Hansen, A. Larsen, H. Norman, E. Andersson, Dynamic airline scheduling: An analysis of the potentials of refleeting and retiming, Journal of Air Transport Management 14 (4) (2008) 163–167.
[11] O. Listes, R. Dekker, A scenario aggregation-based approach for determining a robust airline fleet composition for dynamic capacity allocation, Transportation Science 39 (3) (2005) 367–382.
[12] R. W. Simpson, Computerized schedule construction for an airline transportation system, Tech. rep., M.I.T. Flight Transportation Laboratory (1966).
[13] R. W. Simpson, Scheduling and routing models for airline systems, Tech. rep., M.I.T. Flight Transportation Laboratory (1969).
[14] Y. Chan, Route network improvement in air transportation schedule planning, Tech. Rep. R72-3, M.I.T. Flight Transportation Laboratory (1972).
[15] F. Soumis, J. A. Ferland, J.-M. Rousseau, A model for large scale aircraft routing and scheduling problems, Transportation Research 14B (1980) 191–201.
[16] M. Nikulainen, K. Oy, A simple mathematical method to define demand for schedule planning, AGIFORS Proceedings.
[17] G. Dobson, P. J. Lederer, Airline scheduling and routing in a hub-and-spoke system, Transportation Science 27 (1993) 281–297.
[18] M. Berge, Timetable optimization: Formulation, solution approaches, and computational issues, in: Proceedings of the 34th AGIFORS Annual Symposium, 1994.
[19] R. E. Marsten, r. Subramanian, L. Gibbons, Junior analyst extraordinare (JANE), AGIFORS Proceedings.
[20] B. Rexing, C. Barnhart, T. Kniker, A. Jarrah, N. Krishnamurthy, Airline fleet assignment with time windows, Transportation Science 34 (1) (2000) 1–20.
[21] S. Yan, C.-H. Tseng, A passenger demand model for airline flight scheduling and fleet routing, Computers and Operations Research 29 (11) (2002) 1559–1581.
[22] M. Lohatepanont, C. Barnhart, Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment, Transportation Science 38 (1) (2004) 19–32.
[23] M. Singh, Hub-depeaking techniques at united airlines, in: AGIFORS Schedluling & Strategic Planning Conference, 2006.
[24] T. Jacobs, B. Smith, E. Johnson, Incorporating network flow effects into the airline fleet assignment process, Tech. rep. (2008).
[25] ILOG CPLEX 9.0 User's Manual (2003).