

Chapter 50: Scheduling Problems in the Airline Industry

Xiangtong Qi

Department of Management Science and Information System
The University of Texas at Austin
Austin, TX 78712
qix@mail.utexas.edu

Jian Yang

Department of Industrial and Manufacturing Engineering
New Jersey Institute of Technology
Newark, NJ 07102
yang@adm.njit.edu

Gang Yu

Department of Management Science and Information System
The University of Texas at Austin
Austin, TX 78712
yu@uts.cc.utexas.edu

April 2003

1 Introduction

Being in a time-sensitive and mission-critical business, the airline industry bumps from left to right into all sorts of scheduling problems. To just name a few, it faces the challenges posed by its aircraft scheduling, crew scheduling, scheduling disruption management, and other long-term business planning and real-time operational problems that possess scheduling features.

The scheduling problems faced by the airline industry are even more complicated than the traditional machine scheduling problems. A machine scheduling problem more or less deals with the sequencing and scheduling of a set of jobs to be processed on one or a few machines. On the other hand, an airline has to deal with a set of interwoven complex problems. It has to assign different fleets to thousands of flights under various connectivity and compatibility constraints, then find the route for each individual aircraft so that the just-obtained assignment is fulfilled while the maintenance requirements of the aircraft are met; and at the same time, it has to assign crews to the various flights so that not only the connectivity and compatibility constraints are satisfied but also the rest requirements of the crews and other regulatory constraints are satisfied.

For traditional machine scheduling problems, which can undoubtedly be very difficult, people have at least built up a basic framework under which problems can be described using simple and common terms that reflect the underlying problem settings and assumptions. The research on airline scheduling, on the other hand, is much less standardized. The innate complexity and versatility of airline operations and the relatively young age of the research field make it virtually impossible for researchers, at least at this stage, not to deal with airline scheduling problems in a more piecemeal and *ad hoc* fashion. Having said this, we nevertheless do observe the emergence of unifying trends in the field as more and more common traits are realized for different problems. At this stage, although no unified notational system has appeared, we are already able to classify most problems in the field using several major categories. This chapter aims to give an overview to the categories we deem the most important.

Before going on any further, we make a note on our terminology. Here, an aircraft refers to a physical airplane, a fleet refers to a type of aircraft such as Boeing-737 or MD-80, a flight or flight segment refers to a direct connection from one airport A at a departure time to another airport B at an arrival time, and a crew refers to a person who serves on an aircraft, either a pilot or a flight attendant.

The purpose of this chapter is to provide a review of the up-to-date research on important scheduling problems in the airline industry. Due to the limited space here, we have to strike a balance between the breadth of our coverage and the depth of our analysis. We strive to cover all the major areas of airline scheduling, and emphasize more on modeling and general solution approaches than the in-depth analysis of special-purpose algorithms. For specific technical details, please refer to some of the references listed at the end of the chapter.

There are four major categories of scheduling problems that an airline has to deal with. The first is the aircraft scheduling problem. It is concerned with the assignment of individual aircraft to a pre-determined network of flights to be operated by the airline. Given that different assignments gain different amounts of profit or incur different amounts of cost, the objective is to maximize total profit or minimize total cost under constraints that re-

flect physical feasibilities and governmental regulations such as flight connectivity, aircraft capacity, and maintenance requirements.

The second is the crew scheduling problem. Its goal is to determine the allocation of crew members, namely pilots and flight attendants, to individual flights. Similar to the aircraft scheduling problem, the crew scheduling problem aims to minimize the total allocation cost under various constraints. Moreover, this problem is even more difficult since it involves much more constraints. For example, the problem has to take into account that pilots usually have their preferences to specific flights and all these preferences have to be considered with regard to their seniorities.

The third is the disruption management problem, also called the real-time irregular operations scheduling problem. In daily operations, none of the flight schedule, aircraft schedule, and the crew schedule is likely to be executed without interruptions, due largely to the often occurrences of disruptive events such as bad weather, mechanical failure, and crew sickness. It is therefore imperative to decide in a real-time fashion the best plan to be carried out after the disruptive event has changed the operational environment. At the same time, the airline has also to take into account that the deviation of the new plan from the original plan can be costly and that the new plan must converge back to the original plan after a certain amount of time.

The last category is a combination of airline scheduling problems that can be modeled as traditional machine scheduling problems. The category contains the aircraft landing sequencing problem, the pilot training class scheduling problem, the aircraft scheduling problem with the ground delay program, and the workforce scheduling problem for baggage delivery.

We shall point out that the above four categories have by no means exhausted all airline scheduling problems. For a review of more of these problems, the reader is referred to Yu and Thengvall [74] and Yu and Yang [75].

We organize this chapter as follows. In Section 2, we briefly review some mathematical models that are used in many airline scheduling problems; discuss aircraft scheduling in Section 3 and crew scheduling in Section 4; then address the disruption management problem for airline scheduling in Section 5; finally, we discuss about airline scheduling problems that can be modeled as traditional machine scheduling problems in Section 6.

2 Background on Three Formulations

In this section, we briefly review three combinatorial optimization problems that are extensively used for modeling airline scheduling problems: the set partitioning (covering) problem, the multi-commodity network flow problem, and the Euler tour problem. The first two problems are well known NP-hard problems, while the last one is solvable in polynomial time [27].

2.1 The Set Partitioning Problem

The set partitioning problem can be described as follows. There are a base set $S = \{e_i \mid i = 1, 2, \dots, m\}$ and a collection $\mathcal{S} = \{S_j \mid j = 1, 2, \dots, n\}$ of subsets of S . With each S_j there is associated a cost c_j . For any sub-collection \mathcal{P} of \mathcal{S} , we call it a partition of S when

1) $\cup_{S_j \in \mathcal{P}} S_j = S$, and 2) $S_j \cap S_l = \emptyset$ for any two $S_j, S_l \in \mathcal{P}$. For a partition \mathcal{P} , we define its cost to be $\sum_{S_j \in \mathcal{P}} c_j$, the total cost associated with all the individual subsets in \mathcal{P} . The set partitioning problem is concerned with finding the least costly partition of S . When we relax the second constraint of exclusiveness in the definition for a partition, the resulting sub-collection is called a cover of S and the corresponding problem is called the set covering problem.

There is a natural integer-programming formulation for the set partitioning problem. For any $i = 1, 2, \dots, m, j = 1, 2, \dots, n$, we may use a constant a_{ij} to denote whether or not element e_i is contained in subset S_j : 1 when it is and 0 when it is not. For any $j = 1, 2, \dots, n$, we may use a binary variable x_j to denote whether or not S_j belongs to the solution sub-collection \mathcal{P} : 1 when it is and 0 when it is not. Then, the following integer programming formulation exactly describes the set partitioning problem.

$$\min \sum_{j=1}^n c_j x_j \tag{1}$$

subject to

$$\sum_{j=1}^n a_{ij} x_j = 1, \quad \text{for } i = 1, 2, \dots, m; \tag{2}$$

$$x_j \in \{0, 1\}, \quad \text{for } j = 1, 2, \dots, n. \tag{3}$$

To obtain the formulation for the set covering problem, we only need to change (2) to

$$\sum_{j=1}^n a_{ij} x_j \geq 1, \quad \text{for } i = 1, 2, \dots, m. \tag{4}$$

With respect to airline scheduling problems, usually the base set S is used to model the set of all required flight segments and each subset S_j is used to model a particular subset of connected flight segments that can be consecutively served by an aircraft or a crew. The set partitioning (covering) problem then naturally represents the problem of finding the least costly way to feasibly cover all flight segments with aircraft or crew. Certainly modifications such as the introduction of more constraints will be made when the above formulations are applied to real situations.

Exact methods for solving the set partitioning (covering) problem are usually based on branch and bound algorithms, which need to repeatedly solve various linear programming (LP) relaxation problems. When the LP problems are small, most integer programming solvers will be able to handle the overall problem with ease. When the LP problems become large, however, other techniques need to be employed. One technique people often use is column generation. The technique works as follows. In the beginning, a scaled-down version of the problem with only part of all the columns in the original problem is solved. Then, based on evaluations made on the just-solved problem, some columns are removed from the current problem and some other columns in the original problem are added so that a new LP problem is formed. Next, the new LP problem is solved. The procedure is thus repeated until the current solution is proven to be optimal for the original LP problem.

2.2 The Multi-commodity Network Flow Problem

The network flow model used in airline schedule revolves around the so-called time-space network. In this network, every node stands for a time-location pair representing the departure or landing time and origin or destination airport of a flight. Two types of basic arcs exist in this network: the ground arcs and the flight arcs. A ground arc connects two nodes associated with the same airport and successive time points and usually represents an aircraft or a crew staying at the airport during the corresponding time interval. A flight arc connects two nodes respectively associated with the origin and destination of a flight. We can regard the movements of aircraft or crew members with the various flights as flows in the time-space network. Thus, the problem of finding the cheapest assignment of aircraft or crew members to flights becomes that of finding the minimum-cost integral flows in the network with the observation of constraints such as flow conservation, arc capacity, and others. When we are considering multiple types of aircraft that have different suitabilities with regard to flights, we have to treat the flows corresponding to these different types differentially. Thus, we obviously have to use the multi-commodity version of the network flow problem.

The integer programming formulation of the basic multi-commodity network flow problem can be described as follows. Let there be K commodity types, and the underlying network be $G = (V, I)$, where V is the set of nodes and I the set of arcs. We use (v, v') to denote the arc pointing from node v to node v' . For every $v \in V$, let $IN_v = \{i \mid i = (v', v) \in I \text{ where } v' \in V\}$ be the set of arcs that enter into v and $OUT_v = \{i \mid i = (v, v') \in I \text{ where } v' \in V\}$ be the arcs that leave from v . For $k = 1, 2, \dots, K$ and $i \in I$, let c_{ki} be the cost of sending a unit flow of commodity type k through arc i . For $i \in I$, let u_i be the capacity of arc i . When we use x_{ki} to denote the flow of commodity type k on arc i , our formulation takes the following shape.

$$\min \sum_{k=1}^K \sum_{i \in I} c_{ki} x_{ki} \quad (5)$$

subject to

$$\sum_{i \in IN_v} x_{ki} = \sum_{i \in OUT_v} x_{ki}, \quad \forall k = 1, 2, \dots, K, v \in V; \quad (6)$$

$$\sum_{k=1}^K x_{ki} \leq u_i, \quad \forall i \in I; \quad (7)$$

$$x_{ki} \in \{0, 1, 2, \dots\}, \quad \forall k = 1, 2, \dots, K, i \in I. \quad (8)$$

In the above, (6) enforces flow conservation for each commodity type k at any node v , (7) expresses the capacity constraint for each arc i , and (8) states the integrality constraint for the flow of every commodity on every arc. Note that it is (7) that binds the multiple flow types into one problem: we would only need to solve separate network flow problems were this constraint not present.

The sizes of the problems faced by the airline industry render even the LP relaxations of the corresponding multi-commodity network flow problems difficult to solve. To overcome this difficulty, researchers have developed special tools such as Lagrangian relaxation, column generation, and Dantzig-Wolfe decomposition. The reader may find details in Ahuja, Magnanti, and Orlin [1]. These tools are used for solving the LP relaxations during the

branch and bound or branch and cut processes that eventually solve the original problems to optimality.

2.3 The Euler Tour Problem

A Euler tour in a directed graph is a closed tour along the arcs such that each arc is traversed exactly once, even though some nodes may be traversed multiple times. Given a directed graph, the Euler tour problem seeks such a tour. The aircraft maintenance routing problem where maintenance is always carried out overnight can be modeled as a Euler tour problem. To do so, each node should model an airport and each arc a daily route for a single aircraft, linking its starting airport to its ending airport. Among all airports, some can conduct overnight maintenance checks. When a Euler tour exists, all aircraft can repeatedly experience the same sequence of daily routes though in any given day, they are all assigned to different daily routes.

The sufficient and necessary condition for a directed graph to have a Euler tour is that 1) the graph is connected; and 2) each node has the same out-degree as its in-degree. For an m -arc directed graph satisfying the above two conditions, an $O(m)$ -time algorithm exists for the Euler tour problem. The basic idea of the algorithm is to repeatedly merge arc-disjoint cycles. We start from any node v to traverse previously-untraversed arcs until coming back to v and denote the obtained tour as T . This is achievable due to the two conditions. If all arcs have been traversed, we let T be the Euler tour. Otherwise, we remove from the graph all the arcs in T . There must be a node v' that connects T with the remaining graph. We then as before construct a new tour T' in the remaining graph that starts and ends at v' . Obviously, T and T' can be combined into a single tour that starts and ends at node v . We repeat this process till all arcs are traversed.

3 Aircraft Scheduling

In this section, we introduce the scheduling problem for aircraft, i.e., the assignment of individual aircraft to flights. In practice, the aircraft schedule is determined by a sequence of decision processes. To better understand the problem, we need some knowledge about the processes.

First, the airline must design its time-space flight network comprising the flight segments that it will serve. Each flight segment in the flight network is represented by two nodes and an arc pointing from one of the nodes to the other, with the first node being at the departure time and origin airport of the flight and the second node being at the arrival time and destination airport of the flight. For each airport, the flight network also contains ground arcs that link all neighboring nodes of the same airport in the time-forward fashion. In designing the flight network, many factors have to be considered. These factors include the forecast of market demands, the capacities of the airline's fleets, the competition from other airlines, etc. The main goal is for the network to achieve the maximum profit for the airline. To know how the profit from flying each flight segment is estimated, the reader may refer to Dobson and Lederer [21].

Very often an airline owns many different aircraft types such as Boeing 737, Boeing 777,

DC10, etc. Due to the commonality among aircraft of the same type and the differences in aircraft of different types, airlines usually treat all their same-type aircraft as fleets. The next decision the airline needs to make is then fleet assignment, that of assigning fleet types to the flight segments in its flight network. After this step, the aircraft scheduling problem can be decomposed into separate subproblems under individual fleet types.

In the last step, individual aircraft within each fleet are assigned to the flight segments reserved for the fleet in the fleet assignment stage. This is called the aircraft maintenance routing problem since at this stage, the main concern is to construct a flight schedule for each aircraft so that the aircraft is able to pass the maintenance bases at the frequency mandated by the Federal Aviation Administration (FAA).

In the following, we give an overview of the fleet assignment and aircraft maintenance routing problems. To view an earlier survey, the reader may refer to Gopalan and Talluri [28], which reviewed fleet assignment and aircraft routing along with other topics like traffic forecasting.

3.1 Fleet Assignment

A typical airline conducts its fleet assignment in a periodic fashion, mostly on a daily basis. The factors that influence the assignment of fleet types to flights include passenger demands, seating capacities, operational costs, and various technical and FAA requirements. The objective is either to maximize the total profit or equivalently, minimize the total cost.

3.1.1 The Network Flow Model

The nature of fleet assignment lends itself to a multi-commodity network flow model, where fleet types correspond to commodities, the flight network serves as the underlying network, and assignments correspond to flows. In the following, we introduce the multi-commodity network flow formulation of the fleet assignment problem.

Indices:

k , index for fleet types;

v , index for nodes;

i , index for arcs;

h , index for airports;

Input parameters:

K , number of fleet types;

V , set of all nodes;

I , set of all arcs;

I_F , set of flight arcs;

n_k , number of available aircraft for fleet k ;

H , set of all airports;

s_h , the node associated with airport h and the beginning of the day;

t_h , the node associated with airport h and the end of the day;

c_{ki} , cost of assigning fleet k to flight arc i ;

Decision variables:

x_{ki} , indicating whether fleet type k is assigned to arc i ;

Model:

$$\min \sum_{k=1}^K \sum_{i \in I_F} c_{ki} x_{ki} \quad (9)$$

subject to

$$\sum_{i \in IN_v} x_{ki} = \sum_{i \in OUT_v} x_{ki}, \quad \forall k = 1, 2, \dots, K, v \in V; \quad (10)$$

$$\sum_{k=1}^K x_{ki} = 1, \quad \forall i \in I_F; \quad (11)$$

$$\sum_{h \in H} \sum_{i \in OUT_{s_h}} x_{ki} \leq n_k, \quad k = 1, 2, \dots, K; \quad (12)$$

$$\sum_{i \in OUT_{s_h}} x_{ki} = \sum_{i \in IN_{t_h}} x_{ki}, \quad \forall k = 1, 2, \dots, K, h \in H; \quad (13)$$

$$x_{ki} \in \{0, 1\}, \quad \forall k = 1, 2, \dots, K, i \in I_F; \quad (14)$$

$$x_{ki} \geq 0, \quad \forall k = 1, 2, \dots, K, i \in I. \quad (15)$$

In the model, (9) is the objective function which minimizes the total cost of fleet assignment; (10) states the conservation of flows at each node for each fleet type; (11) makes sure that each flight segment is flown by one and only one aircraft; (12) enforces the resource constraint for each fleet type; (13) is the aircraft balance constraint: the number of aircraft of any particular fleet flying out from any particular airport in the beginning of the day must be the same as the number of aircraft of that fleet flying back to the same airport at the end of the day; (14) is the integral 0-1 constraint on each flight arc; and (15) is the non-negative constraint on each arc.

The above model describes the basic requirement for fleet assignment. To apply the model in practice, however, more details need to be discussed.

Note that it is far from straightforward to estimate each c_{ki} , the cost of assigning fleet type k to flight or ground arc i . In practice, the direct operational costs including fuel costs, crew costs, and landing fees are easy to obtain. The difficulty lies in estimating the indirect costs due to mismatches between the capacities of fleet types and the passenger demands on the flight segments: assigning a bigger aircraft than is needed to a flight induces unnecessarily high direct cost, while assigning a smaller aircraft than is needed to a flight causes potential passengers to be spilled over to overcrowd other flights or be captured by rival airlines or alternative transportation means. Many researchers have tried to find accurate ways to estimate the indirect costs. Barnhart, Kniker, and Lohatepanont [8] proposed the so-called itinerary-based airline fleet assignment, which can capture the assignment costs in more details by combining the basic fleet assignment model and a passenger mix model. Yan and Tsing [68] made similar effort by combining fleet flows and passenger flows in one time-space network.

One important issue in using the above basic model is about the flight connection requirement. An aircraft cannot depart from an airport immediately after it arrives. It has to stay on the ground for some time, say at least 40 minutes. The simplest way to accommodate this requirement is to make a flight arc end in a node at the time when it is ready for its next departure. In practice there are other complex rules regarding flight connection. For

example, the minimum connection time is a function of both incoming and outgoing flight arcs, and hence many connections by the same flight are prohibited. To capture such constraints, the ground arcs have to be refined, for example, by explicitly identifying all possible connection possibilities instead of simply using identical ground arcs. The reader may refer to Rushmeier and Kontogiorgis [50] for more details about this.

In the basic model, the flight segments are given fixed departure and arrival times which were generated in the network design stage. This, of course, may lead to sub-optimal assignments since these times were generated without taking into account the differences in speeds of different fleet types. So it is worthwhile to consider the case where these times can be adjusted at the fleet assignment stage. Actually, we can assume that each flight segment is associated with a flexible departure time window, while the exact departure time within the time window is to be determined during the assignment. We can discretize the time axis to solve the fleet assignment problem with departure time windows: for each discrete time point in the time window of a flight, we create a flight arc departing at that time, with the understanding that one and only one of these flight arcs can have a unit flow. An implementation of such a model may be found in Rexing et al. [49]. Later on, we will show that such a technique is also used in disruption management for flight-aircraft and crew scheduling.

Fleet assignment is mostly done on a daily basis. A weekly schedule can be generated by a repetition of the daily schedule for the weekdays and making proper reductions to the daily schedule for the weekend. A more precise way is to generate a weekly schedule directly while considering different daily requirements. The problem can be modeled in the same way as the daily problem although it is of a much larger size. When departure times are variable within time windows, additional constraints have to be included to ensure that all flights with the same flight number will depart at the same time each day. Such a model was proposed by Ioachim et al. [32].

3.1.2 The Set Partitioning Model

Researchers have also formulated fleet assignment problems as set-partitioning type of problems. We introduce a typical model below (see e.g., Desaulniers et al. [20]).

Let K be the number of fleets, I_F the set of flight segments, and n_k the number of aircraft of fleet $k = 1, 2, \dots, K$. For each fleet k , let \mathcal{S}_k be the collection of subsets of flight segments in I_F that form feasible daily schedules for an aircraft in fleet k . Note that two different \mathcal{S}_k 's may have a nonempty intersection. Let all the subsets in \mathcal{S}_k be $I_{k1}, I_{k2}, \dots, I_{k|\mathcal{S}_k|}$. For any subset I_{kj} in any \mathcal{S}_k , and any flight i in I_F , let a_{ikj} be the binary constant that denotes whether flight i is in subset I_{kj} . Let c_{kj} be the cost of having fleet k cover the subset I_{kj} of flight segments. Also, let x_{kj} be the binary decision variable that designates whether subset I_{kj} in \mathcal{S}_k has been chosen in the fleet assignment solution. In set partitioning terminology, the fleet assignment problem is about choosing subsets from the various \mathcal{S}_k 's to nonredundantly cover all the flights.

Besides the classical set partitioning constraints, we need additional constraints to fully describe our problem. First, the total number of subsets chosen from \mathcal{S}_k cannot exceed the available number n_k of aircraft in fleet k . Secondly, the number of aircraft of any particular fleet flying out from any particular airport in the beginning of the day must be the same as the number of aircraft of that fleet flying back to the same airport at the end of the day. To

deal with this constraint, let H be the set of airports, constant o_{hkj} be 1 if subset I_{kj} starts from airport $h \in H$ and 0 if not, and constant d_{hkj} be 1 if subset I_{kj} ends at h and 0 if not. Now, the fleet assignment problem can be expressed as follows.

$$\min \sum_{k=1}^K \sum_{j=1}^{|\mathcal{S}_k|} c_{kj} x_{kj} \quad (16)$$

subject to

$$\sum_{k=1}^K \sum_{j=1}^{|\mathcal{S}_k|} a_{ikj} x_{kj} = 1, \quad \forall i \in I_F; \quad (17)$$

$$\sum_{j=1}^{|\mathcal{S}_k|} x_{kj} \leq n_k, \quad \forall k = 1, 2, \dots, K; \quad (18)$$

$$\sum_{j=1}^{|\mathcal{S}_k|} o_{hkj} x_{kj} = \sum_{j=1}^{|\mathcal{S}_k|} d_{hkj} x_{kj}, \quad \forall k = 1, 2, \dots, K, h \in H; \quad (19)$$

$$x_{kj} \in \{0, 1\}, \quad \forall k = 1, 2, \dots, K, j = 1, 2, \dots, |\mathcal{S}_k|. \quad (20)$$

3.2 Aircraft Maintenance Routing

After fleet types are assigned to flight segments, the airline then needs to decide how to assign individual aircraft within each fleet to the flight segments to be flown by that fleet. The main concern at this stage is that each aircraft should be guaranteed of sufficiently frequent maintenance checks. Thus the problem is called aircraft maintenance routing.

The relationship between fleet assignment and aircraft maintenance routing can be best illustrated by the following example. Suppose a Boeing 777 fleet has four aircraft, I, II, III, and IV, say, and that the daily flight segments assigned to this fleet are as shown in Table 1.

Table 1: Example of Maintenance Routing

Flight No.	Departure Airport	Departure Time	Arrival Airport	Arrival Time
1	A	8:00am	B	12:00pm
2	B	2:00pm	C	6:00pm
3	C	10:00am	A	3:00pm
4	D	9:00am	B	12:30pm
5	B	2:30pm	E	6:00pm
6	E	11:00am	D	5:00pm

For complete coverage, each of Flights 3(CA) and 6(ED) needs an aircraft; while Flights 1(AB), 2(BC), 4(DB), and 5(BE) together need two aircraft, and there are different possibilities to cover these four flights. One such possibility is to use one aircraft to fly Flights 1 and 2 (ABC) and another to fly Flights 4 and 5 (DBE). Thus we have a set of daily routes for the four aircraft, as might be denoted by $O_1 = \{12(ABC), 3(CA), 45(DBE), 6(ED)\}$. Under

the same notation, $O_2 = \{15(ABE), 3(CA), 42(DBC), 6(ED)\}$ is then another feasible set of daily routes.

Each set of daily routes corresponds to a directed graph where nodes stand for airports and each arc represents a daily route from the starting airport of the day to the ending airport of the day. When a Euler tour or a number of Euler tours (when the graph is disconnected) can be found for the graph, a periodic schedule for the fleet can be derived from the set. For instance, the graph corresponding to O_1 has two Euler tours ACA and DED . We may use Aircraft I (II) to cover route $12(ABC)$ on odd (even) days and route $3(CA)$ on even (odd) days, and use Aircraft III (IV) to cover route $45(DBE)$ on odd (even) days and route $6(ED)$ on even (odd) days. On the other hand, the graph corresponding to O_2 has one Euler tour $AEDCA$. On any given day, each aircraft covers one route in O_2 ; and on the next day, each aircraft is to cover the route that starts from the airport at which its current route ends.

Suppose for every aircraft, an overnight maintenance check is needed once every four days and the check can only be done at airport A. Then in the example, O_1 is clearly not a feasible routing while O_2 is. The aircraft maintenance routing problem for a given fleet is exactly the problem of finding a feasible set of daily routes so that maintenance requirements are met while the flights to be flown by the fleet are already determined.

In reality, there are four major types (Type A, B, C and D) of maintenance checks mandated by the FAA. These checks vary in scope, duration, and the required frequency. In many cases, however, only Type A check, a routine inspection of all major systems of an aircraft including the landing gear, engines and control surfaces, is taken into account when aircraft routing is concerned. This is mainly because Type A check's required frequency, once every 65 flight hours as mandated by the FAA, is much higher than those of all other types of checks. In practice, many airlines take more stringent policies to the effect that this type of checks are called for once every three or four days. Correspondingly, the aircraft routing problems are referred to as the 3-day or 4-day maintenance routing problems.

3.2.1 The Two-Step Approach

The two-step approach applies to the common case where maintenance checks are done overnight. The first step generates a set of daily routes that take care of all flight segments assigned to the given fleet, such as O_1 or O_2 in the preceding example; and the second step generates periodic schedules for individual aircraft that meet all maintenance requirements. These two steps are run repeatedly until the set of daily routes generated by the first step leads to feasible maintenance routes for each aircraft in the second step. The first step usually uses a simple rule like connecting the flights into and out of any airport in a first-in-first-out (FIFO) fashion, which incidentally produces O_1 in the preceding example. The second step, however, can be more involved.

Given the set of airports and the set of daily routes generated by the first step, we can generate a directed graph $G = (N, A)$, such that N corresponds to the airport set and A the route set, with each arc emanating from the starting airport and entering the ending airport of the corresponding route. There is a subset M of N corresponding to airports that can perform maintenance checks. Apparently, $|A|$ aircraft are needed in the fleet to cover all the daily routes. When there is a Euler tour in G (a set of Euler tours in case G has more than one connected components) without any k nodes in $N \setminus M$ in succession, we can construct

feasible periodic routes for all $|A|$ aircraft that meet the maintenance requirements: on any given day, each aircraft covers a route; and on the next day, each aircraft is to cover the route that succeeds its current route on its Euler tour.

For any connected component of a graph, we know that a Euler tour can be found in polynomial time if it exists. The problem now is how to deal with the additional k -node constraint. When $k = 3$, the problem can be solved in polynomial time through a conversion into a Euler tour problem without the k -node constraint. When $k \geq 4$, however, the problem becomes NP-complete. The reader interested in the details can consult Gopalan and Talluri [29] and Talluri [57].

Note that the Euler tour model only deals with the feasibility issue and ignores the cost-optimality issues. In reality, different airports conduct maintenance checks at different prices. If our goal is to find the least expensive aircraft maintenance routing, we should again resort to the multi-commodity network flow model. In the model, each commodity represents one aircraft, and the underlying network is almost the same time-space network studied previously with the exception that, now each pair of airport and calendar day makes up one node and each arc corresponds to one daily route. The problem is to find a path of unit flows for each commodity so that the k -node constraint is satisfied and the total maintenance cost is minimized. This model has the advantage that it provides more opportunity for analyzing maintenance costs and helps the allocation of maintenance bases (see Feo and Bard [24]).

3.2.2 Other Approaches

The two-step approach cannot guarantee the optimality or even feasibility of the solution, because the first step does not consider the maintenance requirements and the second step cannot directly change flight segments within a daily route. It is better if the daily routes can be generated with the maintenance requirements being taken into account. Clarke et al. [16] modeled such a problem as an integer programming problem and solved it using Lagrangian relaxation techniques.

As we have already known, aircraft maintenance routing is usually done fleet by fleet after the occurrence of fleet assignment. Naturally, a more ambitious approach would be to conduct these two activities simultaneously. Barnhart et al. [6] presented such a unified model and solved it using branch and bound and column generation.

There are also researchers who consider more than one type of maintenance checks simultaneously. For example, Sriram and Haghani [53] considered both Type A and Type B checks, where the latter needs to be performed every 300-600 flight hours. The problem was solved by a hybrid heuristic of random and depth-first searches.

4 Crew Scheduling

Crew scheduling deals with the problem of assigning individual crew members to pre-scheduled flights. Like aircraft scheduling, crew scheduling is sequentially divided into two separate stages: crew pairing and monthly crew assignment. We will elaborate on what each of these two stages is about later on. Here, a crew can be either a pilot or a flight attendant. Since each pilot is fleet-associated, pilot scheduling can be decomposed into problems in indi-

vidual fleets. On the other hand, each flight attendant often can work on any fleet. So flight attendant scheduling involves normally all fleets and is a much larger problem. At the same time, pilot scheduling is subject to more strict governmental and contractual regulations, while flight attendant scheduling has to cope with relatively fewer constraints. Besides these differences, similar approaches are usually applied to crew scheduling for both types of crew.

4.1 Crew Pairing

In the crew pairing problem, we are given a set of pre-scheduled flight segments over a period of time. Among all the involved airports, some of them are crew bases where crew members can start and end with over the time period. A legal crew pairing is a sequence of connected flight segments beginning and ending at a crew base that satisfy all legality constraints. The crew pairing problem is that of finding the minimum-cost set of legal crew pairings that cover all the given flight segments.

Both crew pairing and fleet assignment are about covering a given set of flight segments. The crew pairing problem tends to be more difficult mainly because the former involves flights over more than one day while the latter involves flights over one day. Also, crew pairing is usually involved with more constraints due to governmental and contractual restrictions such as maximum daily working hours, minimum overnight rest period, maximum number of flight legs, and maximum time away from a crew base. In addition, a legal crew pairing must start and end at the same crew base, a constraint not to be worried about in fleet assignment.

Much like fleet assignment, existing models for crew pairing can be classified into two main categories, the set partitioning (covering) type and the network flow type. While in fleet assignment, most works are based on the multi-commodity network flow model, crew pairing is more often modeled as a set partitioning (covering) problem. This is mainly due to the difficulty of handling many complex constraints inherent in crew pairing using the network flow model.

4.1.1 The Set Partitioning Model

Using a set partitioning model, each legal pairing is modeled as a subset of flight segments associated with its proper operational cost. The main component of the cost is the so-called pay and credit, defined as the difference between the hours actually needed to cover the subset of flights and the guaranteed hours of pay. The objective of crew pairing is to minimize the total cost of all selected pairings. Let there be m flights and n subsets of legal pairings. Let binary constant a_{ij} indicate whether flight i belongs to subset j , and c_j be the cost associated with subset j . Then, the crew pairing problem can be exactly formulated as a set partitioning problem as described by (1), (2), and (3). If a crew can be on a flight as a passenger, then the corresponding model is of the set covering type, where (2) is replaced by (4).

We can add additional constraints to the above set partitioning or covering model to accommodate more restrictions. For example, suppose airports $1, 2, \dots, B$ are crew bases, there are d_b crew members in airport b , and binary constant f_{bj} indicates whether pairing j starts and ends at airport b . Then, the following constraint addresses the crew availability

issue.

$$\sum_{j=1}^n f_{bj}x_j \leq d_b, \quad b = 1, \dots, B. \quad (21)$$

The major drawback with the set partitioning (covering) model is its huge size. There can be billions of potential legal pairings in a medium size problem. It is infeasible in practice to solve such an integer programming problem to optimality. Naturally, people have been developing various algorithms to achieve near-optimal solutions over the past decades. Most of these algorithms contain primarily two modules, the pairing generation module and the pairing optimization module. The pairing generation module generates legal pairings that can potentially be included in the final solution, and the pairing optimization module solves the current problem based on pairings selected by the pairing generation module. A good solution is supposed to be obtained by running these two modules iteratively.

For crew pairing generation, people often first use a set of randomly generated pairings which is of probably a very high cost. In later iterations, some of the pairings used in past iterations still remain. At the same time, new pairings generated randomly (e.g., Anbil et al. [3] and Klabjan, Johnson, and Nemhauser [36]) or through evaluations (e.g., Graves et al. [30]) enter the pool of generated pairings.

Each pairing optimization problem in the second module is itself a set partitioning problem with less subsets than the original pairing problem. Various methods have been tried on this problem, including the column generation method in Lavoie, Minous, and Odier [38], the branch and price method in Barnhart et al. [7], the Lagrangian relaxation method in Anbil et al. [3], the branch and cut method in Hoffman and Padberg [31], etc. All the afore-mentioned methods have been reported to be successfully implemented and used in real systems.

4.1.2 The Network Flow Model

Network-flow-based models are also used extensively in crew pairing, though still not as much as set-partitioning-based models. Desaulniers et al. [19] formulated the problem as an integer nonlinear multi-commodity network flow problem with additional resource variables. In the model, a commodity models a crew, nodes model airports at different time points, and arcs model various crew activities such as operational flight segments, deadhead flight segments, connections, and rests. Additional resource variables are used to model different crew pairing constraints and regulations. A branch-and-bound algorithm based on an extension of the Dantzig-Wolfe decomposition principle was used to solve this problem. The model has been adopted by Air France. Barnhart and Shenoi [9] arrived to a similar network flow model. They obtained the solution for a relaxed model first and then developed the solution for the original model based on that solution.

In some special cases, the problems may become easy. For example, Yan and Tu [70] proposed a pure network flow model for crew pairing in Taiwan's China Airlines where much fewer constraints are needed. They used the network simplex algorithm to solve the problem. There are also some other approaches that do not belong to conventional mathematical programming techniques, such as the simulated annealing algorithm proposed by Emden-Weinert and Proksch [22] and the genetic algorithms proposed by Levine [41] and Ozdemir and Mohan [46].

4.2 Monthly Crew Assignment

The task following crew pairing is the monthly crew assignment. It assigns individual crew members into trips, i.e., sequences of crew pairings over a certain period of time, usually one month. In practice, two approaches have been taken to tackle the problem. The approach relying on the crew rostering problem tries to find an assignment scheme covering all pairings that is “fair” to all crew members, after taking into account their requirements and preferences. In the other approach called bidline system, each crew member has a different seniority, and the person with a higher seniority has a higher priority to satisfy his/her preference. Crew rostering is more frequently used in Europe, and bidline procedures are more common in North American.

4.2.1 Crew Rostering

The crew rostering problem is often modeled as a set partitioning problem. Here, the pairings form the elements and all possible monthly sequences constitute all the subsets. Ryan [51] used a generalized set partitioning model where each element may need to be covered multiple times to generate a feasible crew rostering with the objective to maximize the total satisfaction of all crew members. The LP relaxation of the problem was first solved by the primal simplex algorithm, then a branch and bound approach was used to generate integer solutions.

Gamache et al. [25] introduced a problem somewhat “dual to” the set partitioning problem, where the number of subsets is fixed and the total cost of the uncovered elements is to be minimized. The problem models the situation where there are a fixed number of crew members and the total cost of the uncovered pairings are to be minimized. They solved the LP relaxation of the problem using column generation, and used heuristics to generate good integer solutions. Local search algorithms are also used in crew rostering problems. Examples include the simulated annealing algorithm by Lucic and Teodorovic [43], and the genetic algorithm by El Moudani et al. [45].

Constraint programming, originally invented in the field of Artificial Intelligence, is widely used for solving crew rostering problems. Dawid, Konig, and Strauss [18] proposed an extended set partitioning model and solved it using a recursive implicit enumeration approach incorporating elements of constraint programming. Caprara et al. [14] and Sellmann et al. [52] used constraint programming as their primary solution techniques.

Some researchers are not content with using one single criterion to measure the fairness of the crew assignment in terms of how satisfied crew members are given their preferences. Instead, they try multi-criterion approaches. Two typical measures are as follows. Given the total flying hours required by flight schedules and the total number of crew members, there is an average flying hour per crew. So one measure of fairness is the deviation of the actual flying time of the trips from the average flying time. Another measure is the deviation of the weekend out-of-home time.

Lucic and Teodorovic [43] studied a multi-criterion problem, and transformed it into a single-criterion one by assigning weights to individual criteria. Teodorovic and Lucic [59] used fuzzy logic models to handle the multiple criteria. El Moudani et al. [45] treated the total cost of the trips as a primary measure and the satisfaction of pilot preferences as a

secondary measure. They first solved the problem minimizing the primary measure while ignoring the secondary objective, and then solved the problem minimizing the secondary measure under the constraint that the primary measure be not worse off than the optimal level obtained previously by a certain percentage.

4.2.2 Bidline Systems

In a bidline system, first trips are generated, and then crew members choose their preferred trips in the order of their seniority. It is natural that some trips are preferable to others due to differences in difficulties, lengths of night flying, time zone crosses, etc. As a consequence, senior crew members always get their ideal duties while junior ones often get unwanted duties. It is therefore up to the trip generation stage to ensure that fairness can still be expected from the system.

Jones [35] was one of the earliest to build a fair bidline system using an algorithmic rather than manual approach. The system uses techniques in expert-system design and other heuristics. Jarrah and Diamond [33] used a set partitioning model to generate trips and solve the problem using column generation techniques. Christou et al. [15] developed a two-phase trip-generation algorithm. The first phase constructs as many high-quality trips as possible, and the second phase uses a genetic algorithm to select the best trips from the pool of trips.

The preferential bidding system combines features of both the crew rostering and bidline approaches. In such a system, crew members with higher seniority still have higher priority to satisfy their preferences. But the preferences are satisfied under the condition that the remaining uncovered trips can still be fully covered by other junior crew members. Gamache, Soumis, and Villeneuve [26] proposed an iterative approach to solve this problem. They determined each crew member's trip one by one from the most senior to the most junior, trying to maximize his/her preference while keeping the remaining problem feasible.

5 Disruption Management for Airline Scheduling

Flight, aircraft, and crew schedules are all generated in advance at an airline's planning stage. When these schedules are being executed, however, various disruptions may occur that render the schedules unexecutable if unchanged. Possible disruptions stem from equipment failure, crew sickness, bad weather, air traffic control restrictions, and so on. Due to their frequent occurrences, the ability to dynamically revise the original schedules to suit the newly changed operational environment after disruptions is very important to the airline. Disruption management refers to this process of plan adjustment.

5.1 An Overview of Disruption Management

We feel the need for an overview of disruption management because this is yet a relatively new area. We shall first base our discussion on a general context rather than the particular airline scheduling setting. Suppose we have an operational plan that is optimal or near optimal under the most expected environment. When the plan is being executed, disruptions

may occur that change the environment abruptly from time to time. As a consequence, the original operational plan may not remain optimal or even feasible. After the occurrence of a disruption, therefore, we need to revise the original plan to make it suitable for the new environment.

In addition to the suitability requirement of the revised new plan for the new operational environment, the new plan should also not be too far away from the original one, since the deviation incurs its own cost in real life. For instance, in the airline scheduling setting, the new flight schedule after a major storm should deviate from the published schedule as less as possible even a certain degree of deviation is inevitable, because any deviation causes confusion and inconvenience to passengers and other operations, real extra operational expenditures, and other penalties. Moreover, in cases where the original plan is to be repeatedly executed, the new plan should gradually converge back to the original plan over a period of time during which no new disruption strikes. Also, the new plan should be generated in a short period of time since it is immediately needed after the occurrence of a disruption.

The study of disruption management originates from airline scheduling and has many applications in other fields such as production planning and scheduling, telecommunications, and public sectors (see e.g., Clausen et al. [17]). Research in airline operations disruption management has brought about its own new concepts and philosophies on handling uncertainties, among which partial solutions and multiple solutions are two important ones.

In the airline setting, it is sometimes impossible to find a new high-quality plan that covers the whole planning horizon and takes all resources and commitments into consideration in a timely fashion. When this is the case, the airline can first settle with a quickly-found partial solution addressing only the most immediate and important decisions, and during the execution of the partial solution, look for a more considerate and longer-term plan that can take over from the partial solution.

The ability of a disruption management system to provide multiple solutions is also important, since some issues arising in real time cannot be addressed by the underlying mathematical model without human intervention. When multiple solutions are presented to the human decision maker, however, he/she will have a better chance to find one that addresses the issues.

5.2 Flight-aircraft Re-scheduling

When a disruption occurs, flights, aircraft, and crew members may all need to be re-scheduled. In practice, this is done sequentially, with flights and aircraft being together re-scheduled first. If no satisfactory new crew schedule can be generated after the new flight-aircraft schedule is generated, then the whole process will have to be repeated. We discuss disruption management for flight-aircraft scheduling first.

Two important issues of the flight-aircraft re-scheduling problem are related to the original flight-aircraft schedule. First, the disruption management problem often has a time window. It is required that the new schedule must converge back to the original schedule after a predetermined time window to mitigate the long-term effect of the disruption. Specifically, all aircraft must be at correct airports by the end of the time window. Second, within the time window, it is also preferred that the new schedule deviate from the original schedule as minimally as possible. We will explain how to measure and achieve this goal later.

To re-schedule the flights and aircraft together, there are several options with different deviation costs, i.e., cost differences between the new and original actions. These options include delaying some flights by certain amounts of time (still to be covered by the same aircraft as originally planned), canceling some flights, using allowable types of aircraft other than the original types to cover flights, and ferrying in aircraft from other airports to cover flights. The objective of the disruption management problem is to find a new flight-aircraft schedule with the minimum total deviation cost that converge back to the original schedule within the time window.

The problem can be modeled as a multi-commodity network flow problem revolving around an underlying time-space network similar to the one used in fleet assignment. Each commodity represents a type of aircraft. The network represents the original flight schedule, where ground arcs represent aircraft waiting on the ground and flight arcs represent flights. Aircraft-flight assignments are modeled as flows on the arcs. The time horizon of the network spans from the moment right after the occurrence of the disruption to the time when the time window ends. Every node representing an airport at the beginning of the time horizon has a given integer amount of inflow. This puts the number of aircraft available at the airport right after the disruption into the model. Also, every node representing an airport at the end of the time horizon is required to have a certain integer amount of inflow. This forces the aircraft to reach the positions for resuming the original schedule after the time window. Multiple delay arcs parallel to each flight arc, arcs that connect the same set of airports at delayed time points, are generated, so that having a unit flow at one and only one of these arcs can represent the flight being delayed. Having a zero flow on a flight arc and all its delay arcs naturally represent the flight being canceled. Ferry arcs, very similar to flight arcs, can be added to the network so that a unit flow on a ferry arcs represent an aircraft being ferried from one airport to another.

Besides flight delays and cancellations, another source of huge deviation cost comes from having to use different aircraft to cover different flights that are originally covered by a single aircraft. Passengers have strong preferences to stay in the same aircraft. A model that penalizes this kind of deviation can use a modeling apparatus called protection arcs. For two connected flights that originally use the same aircraft, a protection arc starts from the starting node of the first flight and ends at the terminal node of the second flight. A unit flow on the protection arc means that both flights are covered by the same aircraft. When this arc is assigned a cost that is less than the total costs of the two flight arcs, there will be incentive to use the same aircraft for both flights.

Teodorovic and Guberinic [58] used the network flow model to study the flight-aircraft re-scheduling problem. But they only considered the option of delaying flights. Teodorovic and Stojkovic [60] extended the preceding work to include flight cancellations. Jarrah et al. [34] presented two special cases based on the network flow model, one considering only flight delays and the other considering only flight cancellations. While delays and cancellations were not addressed simultaneously, this approach was already considered practical enough to be implemented by United Airlines (see Rakshit, Krishnamurthy, and Yu [48]). Yan and Yang [71] were the first to incorporate flight delays, cancellations, and ferryings in a single model. Yan and Lin [67] extended the above model to handle airport closures. Yan and Tu [69] extended the same model to tackle multiple fleet substitutions. Other related works include Arguello, Bard, and Yu [4, 5] and Cao and Kanafani [12, 13].

Thengvall, Bard, and Yu [62] first introduced the protection arcs to reduce the deviation cost due to passenger dissatisfaction over having to change aircraft in connection airports. Thengvall, Yu, and Bard [63] considered probably the most complete model so far. They considered flight delays, cancellations, the above peculiar deviation, aircraft ferrying, fleet substitution, and hub closures. Recently, Stojkovic et al. [54] studied a special case of the problem where the disruption is small enough for the airline to be able to keep the original aircraft itineraries. The problem is interesting in that it can be modeled by a pure network flow model and thus solved in polynomial time.

5.3 Crew Re-scheduling

The next step after flight-aircraft re-scheduling is to revise the crew schedule with respect to the new flight-aircraft schedule. Recall that in crew scheduling, crew pairings lasting 2-5 days are generated to cover all flights, and crew members are assigned to the pairings through bidding or rostering. When the flight schedule is changed, some original crew pairings are broken. Some disruptions, such as crew sickness and emergency leaves, impair the execution of the original crew schedule directly rather than through a changed flight-aircraft schedule.

The goal of crew re-scheduling anyhow is to repair the broken pairings so that the entire system can return to the original schedule efficiently within a given time window. The options to be used for the repairing include forming new crew pairings, breaking up old crew pairings, using reserved crew members, and crew deadhead (a crew travels as a passenger to an airport and joins a pairing or back to the home base).

Relative to that of flight-aircraft re-scheduling, the crew re-scheduling literature is rather sparse. Teodorovic and Stojkovic [61] used a FIFO rule to assign crew members to new flight-aircraft schedules. Wei, Yu, and Song [66] developed a heuristic-based search algorithm for the problem. The solution approach was successfully implemented for Continental Airlines (see Yu et al. [72]). Lettovsky, Johnson, and Nemhauser [40] proposed an integer programming model for the problem, and used a primal-dual subproblem simplex method to solve its LP relaxations. Computational results showed that medium size problems can be solved to optimality within minutes.

Traditionally, people dealt with the flight-aircraft and crew re-scheduling problems sequentially. Recently, however, Stojkovic and Soumis [55] made an attempt to consider the two problems simultaneously under a single framework. In the model, changes to the existing flight and crew schedules are considered simultaneously, while the planned aircraft itineraries are kept intact. The objective is to minimize the total cost due to flight delays and cancellations and crew schedule changes. The problem was formulated as an integer non-linear multi-commodity network flow problem with time windows and additional constraints. The solution approach used Dantzig-Wolfe decomposition combined with branch-and-bound. The approach was tested on several input data sets. All of them were successfully solved very quickly.

6 Problems Modeled as Machine Scheduling Problems

In this section, we introduce several airline scheduling problems that can be modeled as machine scheduling problems. These problems can be described by the machine scheduling terminology in terms of jobs, machines, job processing times, job completion times, job due dates, etc. Currently, these problems have not received enough attention from the machine scheduling research community, and yet, they are inspired by real applications and pose new opportunities.

6.1 Scheduling of Aircraft Landings

We start with the problem of scheduling aircraft landings at an airport. There are n aircraft flying toward an airport during a planning cycle. Without other competitors for the same runway, each aircraft has an ideal landing time d_i which may be its published landing time or the landing time resulting from following the most fuel-efficient flying speed. On the other hand, each aircraft i has its earliest possible landing time a_i resulting from flying at its top speed, and its latest possible landing time b_i resulting from consuming all the fuel it carries before landing. Aircraft i must land within the time window $[a_i, b_i]$. There is also a corresponding cost associated with the aircraft's real landing time C_i : an α_i per unit time penalty for being earlier than d_i and a β_i per unit time penalty for being later than d_i . When all the n aircraft are competing for the same runway, they still have to observe a certain forbidding-period rule: right after aircraft i has landed, there is a p_i amount of time during which no aircraft can land. The job of the aircraft landing scheduling problem is to find a landing time for each of the n aircraft within its time window such that no aircraft lands within a preceding aircraft's forbidding period and the total cost associated with these time spots is the minimum possible.

A single machine scheduling problem with time windows and the earliness-tardiness objective can be used to model this problem. Each aircraft can be modeled as a job and the runway the single machine. The ideal landing time d_i now corresponds to job i 's due date, and $[a_i, b_i]$ the job's time window into which its completion time must fall. Aircraft i 's forbidding period p_i is now the processing time required by job i . Job i 's completion time C_i in turn corresponds to aircraft i 's landing time. Let binary variable x_{ki} indicate whether the k th job in the schedule is job i . Then, the machine scheduling problem that exactly addresses concerns of the aircraft landing scheduling problem can be formulated as in the following.

$$\min \sum_{i=1}^n (\alpha_i \max\{d_i - C_i, 0\} + \beta_i \max\{C_i - d_i, 0\}) \quad (22)$$

subject to

$$\sum_{i=1}^n x_{ki} = 1, \quad \forall k = 1, 2, \dots, n; \quad (23)$$

$$\sum_{k=1}^n x_{ki} = 1, \quad \forall i = 1, 2, \dots, n; \quad (24)$$

$$\sum_{i=1}^n x_{k+1,i} C_i \geq \sum_{i=1}^n x_{ki} (C_i + p_i), \quad \forall k = 1, 2, \dots, n-1; \quad (25)$$

$$a_i \leq C_i \leq b_i, \quad \forall i = 1, 2, \dots, n. \quad (26)$$

In the formulation, the objective (22) is clearly to minimize the total earliness and tardiness cost; (23) states that one and only one job occupies each position and (24) states that each job occupies one and only one position; (25) expresses each job's required processing time; and (26) enforces the hard time window for each job.

While earliness-tardiness scheduling problems have been extensively studied in the past two decades, to our knowledge no research has been conducted on the above problem in the machine scheduling context. In airline scheduling, people have been using mathematical programming techniques to address this kind of problems. Beasley et al. [10] presented a mixed-integer zero-one formulation for the problem and solved it using a LP relaxation tree search algorithm and a heuristic algorithm. Ernst, Krishnamurthy, and Storer [23] worked on the same model and developed a specialized simplex algorithm for the problem.

6.2 Class Scheduling for Pilot Training

This section introduces the class scheduling problem at the training center of a major airline. The airline's pilots are often awarded with new positions which require them to receive additional trainings on new skills at the airline's training center. A major problem emerging from this need for pilot training is the so-called training class scheduling problem: with limited training device availabilities, how to determine the daily activities of all pilots in the training center so that they can complete the training in as short amounts of time as possible (Yu, Dugan, and Arguello [73]).

Specifically, we are given n classes to be scheduled. Each class has a sequential schedule to have the occasion to use certain training devices. Also, some devices can substitute for other devices. On the other hand, the supply of the training devices is limited and the classes have to use the commonly-demanded devices on different days. An unlimited number of free days can be inserted into the schedule of each class when the devices needed are being used by other classes. The inserted days certainly lengthen the duration of each class, which is undesirable since each pilot being trained in a class remains unproductive for its entire duration. The objective of the class scheduling problem is therefore to dispatch the training devices to classes in such a way that the total weighted completion time of the classes is minimized.

The problem can again be modeled as a machine scheduling problem. Here, each training device is modeled as a machine, each class as a job, and any day's activity of a class as a unit-processing-time operation. In the machine scheduling literature, there are some existing works that handle some aspects of the class scheduling problem. For instance, Linn and Zhang [42] considered a hybrid flow shop problem, where a hybrid flow shop has several stations each of which has multiple machines. A job needs to be processed sequentially through these stations and only needs one of the machines at any station. Brucker, Jurisch, and Kramer [11] studied a multi-purpose machine scheduling problem, where different machines are partially substitutable, i.e., an operation of a job can be done by any one of a set of candidate machines. Lee [39] considered a scheduling problem with machine unavailability,

where a machine may have variable capability over time. On the other hand, none of the existing work in machine scheduling describes all aspect of the class scheduling problem.

Qi, Bard, and Yu [47] gave a full account of the class scheduling problem and solved it using a branch and bound algorithm. In the method, each node D in the branching tree that is t levels down from the root of the tree corresponds to a partial schedule for all classes from days 1 to t , which describes for each class its activities during this period. Each of node D 's successor node in the tree corresponds to a $(t + 1)$ -day partial schedule for all the classes whose activities during the first t days coincide with those corresponding to D . Various elimination rules and effective lower bounds were used to cut unpromising branches to accelerate the searching speed.

As the problem size grows, however, the running time of the branch and bound algorithm increases fairly rapidly. A faster rolling horizon heuristic was then developed. In this approach, classes are ordered in a series as Class 1, 2, \dots , n , and the problem is solved iteratively with each iteration a problem with a smaller size of h classes being solved. In the first iteration, Classes 1, 2 to h are scheduled to their completions; in the second iteration, Class 1's schedule is held unchanged, and Classes 2, 3 to $h + 1$ are scheduled to their completions, with the new schedules for Classes 2,3 to h possibly being different from the ones found in the last iteration; this process keeps on going until in the $(n - h + 1)$ th iteration, Classes $(n - h + 1)$, $(n - h + 2)$ to n are all scheduled to their completions. Each iteration is in turn solved by the branch and bound algorithm.

6.3 Scheduling with the Ground Delay Program

The Ground Delay Program (GDP) is one of the several programs that the FAA is currently administering for more efficient and equitable use of the airspace and airports. When bad weather develops around an airport and reduces its operational capacity, the FAA may initiate the GDP which restricts any given airline's landing to the airport to a few time slots. These time slots are usually later than the planned landing times for the airline's flights into the airport. So it is important for the airline to re-schedule their flights so that a certain measure of delay can be minimized. Note that this problem also falls into the category of disruption management. We present it here because of its use of machine scheduling modeling.

The problem faced by an airline at an airport with the GDP is stated as follows. There is a set I of in-flights and a set K of out-flights at the airport. An out-flight needs resources, such as crew members and connecting passengers, from some of the in-flights. So the out-flight can not depart until the corresponding in-flights have landed. Therefore, how much each out-flight is to be delayed depends on how the time slots are assigned to the in-flights. Given a planned departure schedule of all out-flights, the scheduling problem with the GDP is to assign the time slots to in-flights so that a certain measure of lateness related to the out-flight departures is minimized. For certain measures of lateness, we can model the airport as a single machine with a limited set of time slots and model the in-flights as jobs with due dates. Thus the problem becomes a due-date-related machine scheduling problem.

When the maximum lateness of the out-flights is to be minimized, we can in the machine scheduling model define the due date of a job corresponding to an in-flight to be the earliest planned departure time of the out-flight that need resources from the in-flight. Using a

pairwise exchange argument, it can be shown that the earliest due-date (EDD) rule solves the problem to optimality. When the number of late out-flights is to be minimized, we can still define the job due dates in the same fashion. This time, however, the corresponding machine scheduling problem is strongly NP-hard. A heuristic has been proposed for the problem. For detailed discussions about scheduling with the GDP, the reader is referred to Vasquez-Marquez [65] and Luo and Yu [44].

6.4 Scheduling with Varying Machine Speeds

Amaddeo, Nawijn, and van Harten [2] studied a baggage handling problem where the handling speed varies under the varying worker availabilities and different baggages are allowed to be handled simultaneously. The problem can be modeled as a single-machine scheduling problem in which the machine speed varies over time, several jobs might occupy the machine simultaneously, and the goal is to minimize the total weighted completion time of the jobs. More specifically, there are n jobs to be processed on a machine; the total amount of work required by job i is q_i and its weight is w_i ; the machine speed at time t is given at $m(t)$; the machine is to dedicate a partial speed $m_i(t)$ to job i at time t , under the constraint that $m(t) = \sum_{i=1}^n m_i(t)$; the completion time C_i of job i is determined by the equation $q_i = \int_0^{C_i} m_i(t) dt$; and the goal is to find the machine allocation schedule $\{m_i(t) \mid i = 1, 2, \dots, n, t \geq 0\}$ that minimizes the total weighted completion time $\sum_{i=1}^n w_i C_i$ of the jobs.

It has been shown that there exists an optimal schedule in which no jobs are processed concurrently and no preemption ever occurs. Based on this property, a conventional branch and bound algorithm was used to solve the problem in which a node in the searching tree represents a partial schedule where only some of the jobs are scheduled. The problem itself has been proved to be NP-hard. A similar problem has also been studied by Surkis and Dogramaci [56], who proposed a simple heuristic algorithm. Note that the above problem is different from the variable-speed machine scheduling problem studied by Trick [64]. In the latter, the machine speeds can be adjusted at varying prices.

References

- [1] Ahuja, R.K., Magnanti, T.L., and Orlin, J.B., *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, New Jersey, 1993.
- [2] Amaddeo, H.F., Nawijn, W.M., and van Harten, A., One-machine job-scheduling with non-constant capacity - Minimizing weighted completion times, *European Journal of Operational Research*, 102, 502-512, 1997.
- [3] Anbil, R. et al., Recent advances in crew-pairing optimization at American Airlines, *Interfaces*, 21:1, 62-74, 1991.
- [4] Arguello, M.F., Bard, J.F., and Yu, G., A GRASP for aircraft routing in response to groundings and delays. *Journal of Combinatorial Optimization*, 5, 211-228, 1997.

- [5] Arguello, M.F., Bard, J.F., and Yu, G., Models and methods for managing airline irregular operations aircraft routing. In *Operations Research in the Airline Industry*, Yu, G. Ed., Kluwer Academic Publishers, Boston, 1-45, 1997.
- [6] Barnhart, C. et al., Flight string models for aircraft fleeting and routing, *Transportation Science*, 32, 208-220 1998.
- [7] Barnhart, C. et al., Branch-and-price: Column generation for solving huge integer programs, *Operations Research*, 46, 316-329, 1998.
- [8] Barnhart, C., Kniker, T.S., and Lohatepanont, M., Itinerary-based airline fleet assignment, *Transportation Science*, 36, 199-217, 2002.
- [9] Barnhart, C. and Shenoi, R.G., An approximate model and solution approach for the long-haul crew pairing problem, *Transportation Science*, 32, 221-231, 1998.
- [10] Beasley, J.E. et al., Scheduling aircraft landings - The static case, *Transportation Science*, 34, 180-197, 2000.
- [11] Brucker, P., Jurisch, B., and Kramer, A., Complexity of scheduling problems with multi-purpose machines, *Annals of Operations Research*, 70, 57-73, 1997.
- [12] Cao, J. and Kanafani, A., Real-time decision support for integration of airline flight cancellations and delays, part I: mathematical formulations, *Transportation Planning and Technology*, 20, 183-199, 1997.
- [13] Cao, J. and Kanafani, A., Real-time decision support for integration of airline flight cancellations and delays, part II: algorithms and computational experiments, *Transportation Planning and Technology*, 20, 201-217, 1997.
- [14] Caprara, A. et al., Integrating constraint logic programming and operations research techniques for the crew rostering problem, *Software-Practice & Experience*, 28, 49-76, 1998.
- [15] Christou, I.T. et al., A two-phase genetic algorithm for large-scale bidline-generation problems at Delta Airlines, *Interfaces*, 29(5), 51-65, 1999.
- [16] Clarke, L. et al., The aircraft rotation problem, *Annals of Operations Research*, 69, 33-46, 1997.
- [17] Clausen, J. et al., Disruption management. *ORMS Today*, 28, 40-43, 2001.
- [18] Dawid, H., Konig, J., and Strauss, C., An enhanced rostering model for airline crews, *Computers & Operations Research*, 28, 671-688, 2001.
- [19] Desaulniers, G. et al., Crew pairing at Air France, *European Journal of Operational Research*, 97, 245-259, 1997.
- [20] Desaulniers, G. et al., Daily aircraft routing and scheduling, *Management Science*, 43, 841-855, 1997.

- [21] Dobson, G. and Lederer, P.J., Airline scheduling and routing in a hub-and-spoke system, *Transportation Science*, 27, 281-297, 1993.
- [22] Emden-Weinert, T. and Proksch, M., Best practice simulated annealing for the airline crew scheduling problem, *Journal of Heuristics*, 5, 419-436, 1999.
- [23] Ernst, A.T., Krishnamurthy, M., and Storer, R.H., Heuristic and exact algorithms for scheduling aircraft landings, *Networks*, 34, 229-241, 1999.
- [24] Feo, T.A. and Bard, J.F., Flight scheduling and maintenance base planning, *Management Science*, 35, 1415-1432, 1989.
- [25] Gamache, M. et al., A column generation approach for large-scale aircrew rostering problems, *Operations Research*, 47, 247-263, 1999.
- [26] Gamache, M., Soumis, F., and Villeneuve, D., The preferential bidding system at Air Canada, *Transportation Science*, 32, 246-255, 1998.
- [27] Garey, M.R. and D.S. Johnson (1979), *Computers and Intractability*, W.H. Freeman, San Francisco.
- [28] Gopalan, R. and Talluri, K.T., Mathematical models in airline schedule planning: A survey. *Annals of Operations Research*, 76, 155-185, 1998,
- [29] Gopalan, R. and Talluri, K.T., The aircraft maintenance routing problem, *Operations Research*, 46, 260-271, 1998.
- [30] Graves, G.W. et al., Flight crew scheduling, *Management Science*, 39, 736-745, 1993.
- [31] Hoffman, K.L. and Padberg, M., Solving airline crew scheduling problems by branch-and-cut, *Management Science*, 39, 657-682, 1993.
- [32] Ioachim, I. et al., Fleet assignment and routing with schedule synchronization constraints, *European Journal of Operational Research*, 119, 75-90, 1999.
- [33] Jarrah, A.I.Z. and Diamond, J.T., The problem of generating crew bidlines, *Interfaces*, 27(4), 49-64, 1997.
- [34] Jarrah, A.I.Z. et al., A decision support framework for airline flight cancellations and delays, *Transportation Science*, 27, 266-280.
- [35] Jones, R.D., Development of an automated airline crew bid generation system, *Interfaces*, 19(4), 44-51, 1989.
- [36] Klabjan, D., Johnson, E.L., and Nemhauser, G.L., Solving large airline crew scheduling problems: Random pairing generation and strong branching, *Computational Optimization and Applications*, 20, 739-751, 2001.
- [37] Klabjan, D. et al., Airline crew scheduling with regularity, *Transportation Science*, 35, 359-374, 2001.

- [38] Lavoie, S., Minous, M., and Odier, E., A new approach for crew pairing problem by column generation with an application to air transportation, *European Journal of Operational Research*, 35, 45-58, 1988.
- [39] Lee, C.-Y., Two-machine flowshop scheduling with availability constraints, *European Journal of Operational Research*, 114, 420-429, 1999.
- [40] Lettovsky, L., Johnson, E.L., and Nemhauser, G.L., Airline crew recovery, *Transportation Science*, 34, 337-348, 2000.
- [41] Levine, D., Application of a hybrid genetic algorithm to airline crew scheduling, *Computers & Operations Research*, 23, 547-558, 1996.
- [42] Linn, R. and Zhang, W., Hybrid flow shop scheduling: A survey, *Computers & Industrial Engineering*, 37, 57-61, 1999.
- [43] Lucic, P. and Teodorovic, D., Simulated annealing for the multi-objective aircrew rostering problem, *Transportation Research Part A: Policy and Practice*, 33, 19-45, 1999.
- [44] Luo, S.J. and Yu, G., On the airline schedule perturbation problem caused by the ground delay program, *Transportation Science*, 31, 298-311, 1997.
- [45] El Moudani, W. et al., A bi-criterion approach for the airlines crew rostering problem, *Lecture Notes in Computer Science*, 1993, 486-500, 2001.
- [46] Ozdemir, H.T. and Mohan, C.K., Flight graph based genetic algorithm for crew scheduling in airlines, *Information Science*, 133, 165-173, 2001.
- [47] Qi, X., Bard, J.F., and Yu, G., Class scheduling for pilot training, *Operations Research*, forthcoming, 2003.
- [48] Rakshit, A., Krishnamurthy, N., and Yu, G., Systems operations advisor: A real-time decision support system for managing airline operations at United Airlines, *Interfaces*, 26, 50-58, 1996.
- [49] Rexing, B. et al., Airline fleet assignment with time windows, *Transportation Science*, 34, 1-20, 2000.
- [50] Rushmeier, R.A. and Kontogiorgis, S.A., Advances in the optimization of airline fleet assignment, *Transportation Science*, 31, 159-169, 1997.
- [51] Ryan, D.M., The solution of massive generalized set partitioning problems in aircrew rostering, *The Journal of the Operational Research Society*, 43, 459-467, 1992.
- [52] Sellmann, M. et al., Crew assignment via constraint programming: Integrating column generation and heuristic tree search, *Annals of Operations Research*, 115, 207-225, 2002.

- [53] Sriram, C. and Haghani, A., An optimization model for aircraft maintenance scheduling and re-assignment, *Transportation Research Part A: Policy and Practice*, 37, 29-48, 2003.
- [54] Stojkovic, G. et al., An optimization model for a real-time flight scheduling problem, *Transportation Research Part A: Policy and Practice*, 36, 779-788, 2002.
- [55] Stojkovic, G, and Soumis, F., An optimization model for the simultaneous operational flight and pilot scheduling problem, *Management Science*, 47, 1290-1305, 2001.
- [56] Surkis, J. and Dogramaci, A., Minimizing the sum of weighted completion times of n-independent jobs when resource availability varies over time: Performance of simple priority rule, *Naval Research Logistics Quarterly*, 35, 35-47, 1988.
- [57] Talluri, K.T., The four-day aircraft maintenance routing problem, *Transportation Science*, 32, 43-53, 1998.
- [58] Teodorovic, D. and Guberinic, S., Optimal dispatching strategy on an airline network after a schedule perturbation, *European Journal of Operational Research*, 15, 178-182, 1984.
- [59] Teodorovic, D. and Lucic, P., A fuzzy set theory approach to the aircrew rostering problem, *Fuzzy Sets and Systems*, 95, 261-271, 1998.
- [60] Teodorovic, D. and Stojkovic, G., Model for operational daily airline scheduling. *Transportation Planning and Technology*, 14, 273-285, 1990.
- [61] Teodorovic, D. and Stojkovic, G., Model to reduce airline schedule disturbances, *Journal of Transportation Engineering*, 121, 324-331, 1995.
- [62] Thengvall, B.G., Bard, J.F., and Yu, G., Balancing user preferences for aircraft schedule recovery during irregular operations, *IIE Transactions on Operations Engineering*, 32, 181-193, 2000.
- [63] Thengvall, B.G., Yu, G., and Bard, J.F., Multiple fleet aircraft schedule recovery following hub closures, *Transportation Research Part A: Policy and Practice*, 35, 289-308, 2001.
- [64] Trick, M.A., Scheduling multiple variable-speed machines, *Operations Research*, 42, 234-248, 1994.
- [65] Vasquez-Marquez, A., American-Airlines arrival slot allocation system (ASAS), *Interfaces*, 21(1), 42-61, 1991.
- [66] Wei, G., Yu, G., and Song, M., Optimization model and algorithm for crew management during airline irregular operations, *Journal of Combinatorial Optimization*, 1, 305-321, 1997.
- [67] Yan, S. and Lin, C., Airline scheduling for the temporary closure of airports, *Transportation Science*, 31, 72-82, 1997.

- [68] Yan, S. and Tsing, C.-H., A passenger demand model for airline scheduling and routing, *Computers & Operations Research*, 29, 1559-1581, 2002.
- [69] Yan, S. and Tu, Y.-P., Multifleet routing and multistop flight scheduling for schedule perturbation, *European Journal of Operational Research*, 103, 155-169, 1997.
- [70] Yan, S. and Tu, Y.-P., A network model for airline cabin crew scheduling, *European Journal of Operational Research*, 140, 531-540, 2002.
- [71] Yan, S. and Yang, D., A decision support framework for handling schedule perturbations, *Transportation Research Part B: Methodology*, 30, 405-419, 1996.
- [72] Yu, G. et al., A new era for crew recovery at Continental Airlines, *Interfaces*, 33(1), 5-22, 2003.
- [73] Yu, G., Dugan, S., and Arguello, M., Moving toward an integrated decision support system for manpower planning at Continental Airlines: Optimization of pilot training assignments. in *Industrial Applications of Combinatorial Optimization*, Yu, G., Ed., Kluwer Academic Publishers, Boston, 1-24, 1998.
- [74] Yu, G. and Thengvall, B., Optimization in the airline industry, in *Handbook of Applied Optimization*, Pardalos, P.M. and Resende, M.G.C., Eds., Oxford University Press, New York, 2002.
- [75] Yu, G. and Yang, J., Optimization applications in air transportation, in *A Handbook for Combinatorial Optimization*, Du, D.-Z. and Pardalos, P.M., Eds., Kluwer Academic Publishers, 635-726, 1999.