

Aircraft and crew scheduling for fractional ownership programs

Wei Yang · Itr Z. Karaesmen · Pinar Keskinocak · Sridhar Tayur

Published online: 4 December 2007
© Springer Science+Business Media, LLC 2007

Abstract Fractional aircraft ownership programs, where individuals or corporations own a fraction of an aircraft, have revolutionized the corporate aviation industry. Fractional management companies (FMC) manage all aspects of aircraft operations enabling the owners to enjoy the benefits of private aviation without the associated responsibilities. We describe here the development of a scheduling decision support tool for a leading FMC. We present mathematical models, exact and heuristic solution methods. Our computational results using real and randomly generated data indicate that these models are quite effective in finding optimal or near-optimal solutions. The first phase of the implementation of one of these models at the FMC led to a significant improvement in effective utilization of the aircraft, reduction of costs due to reduced empty moves, and hence increased profits.

W. Yang
Management Department, School of Business, Long Island University at C.W. Post, Brookville, NY
11548, USA
e-mail: wei.yang@liu.edu

W. Yang
Chinese-American Center for Service Operations, Southwest Jiaotong University, Cheng Du 610031,
China

I.Z. Karaesmen (✉)
R.H. Smith School of Business, University of Maryland, 4357 Van Munching Hall, College Park, MD
20742, USA
e-mail: ikaraes@rhsmith.umd.edu

P. Keskinocak
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology,
Atlanta, GA 30332, USA
e-mail: pinar@isye.gatech.edu

S. Tayur
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: stayur@andrew.cmu.edu

Keywords Aircraft and crew scheduling · Time-sharing of corporate jets · Decision-support systems · Network flows · Column generation · Branch-and-price

1 Introduction

The general downturn in the US economy and increasing fuel costs since 2001 have brought many passenger airlines to the brink of bankruptcy. Despite the troubled times for air travel, one particular segment of the aviation industry has been growing: the fractional ownership programs. These programs have revolutionized business aviation and created an exciting and rapidly growing market with a wide variety of customers, including celebrities, athletes, small and mid-sized companies who need business travels but cannot afford an entire aircraft. Some large corporations also supplement their own corporate jet fleets with additional flight hours from fractional management companies (FMCs). Since the inception of the time-sharing or fractional ownership business twenty years ago, the number of companies or individuals using these services has increased significantly, from only 3 owners in 1986 to 6217 in 2004 (NBAA 2004). By 2005, the combined fleet of the three major FMCs, NetJets, Flight Options, and Flexjet reached 1000 (Sweetman 2005). According to industry forecasts, the total number of share owners is expected to reach 7000 and the fleet size 1200 by the end of 2007 (NBAA 2004).

Travelers are often frustrated by commercial flights because of delays and inflexible schedules. For those looking for the convenience of a private jet without being responsible for the aircraft, the crew, or other aspects of aircraft operations, one option is to purchase a share of an aircraft from an FMC (Keskinocak 1999). Under this concept, a 1/16 share entitles a customer to 50 hours of annual flight time and a 1/2 share licenses 400 hours (Sweetman 2005). Requested trips are guaranteed within as short as 4 hours of notice before the departure time with access to about 5300 airports in the U.S. as opposed to only 558 served by scheduled carriers (NBAA 2004). This convenience and flexibility come at a price: In 2001 a 1/8 share of a Boeing Business Jet was sold for \$6 million one-time ownership fee, plus \$41,480 monthly management fee, and \$4,360 hourly occupancy fee (Sheridan 2002).

Some regard FMCs as “unscheduled airlines” where the biggest challenge is orchestrating the operational requirements, including aircraft/crew and maintenance scheduling, for utmost customer satisfaction. Hence, for effectively and efficiently managing the operations of the rapidly growing fractional ownership programs, FMCs need to utilize computers and automated tools. In this paper, we describe the models and methods developed to support the daily scheduling decisions of a major FMC. The paper follows the time line of our implementation at one of the largest FMCs, which does not have an internal Operations Research (OR) group. Many of the practitioners at this FMC were initially skeptical of a decision support tool that relied on OR, largely due to their lack of experience with such tools. For this reason, we decided to break the project into two phases. In the first phase, a model and a solution technique was implemented rapidly at the production level leading to significant improvements. As the schedulers became familiar with the tool, and its value could be quantified, the skepticism was replaced with enthusiasm for more advanced features. At the same time, the FMC merged with another to double its size and the need for automated decisions surged. The success of phase 1 has led to phase 2 analysis.

The paper is organized as follows: Sect. 2 describes the challenges in aircraft and crew management and provides a review of the literature. Section 3 introduces the decision support tool. Section 4 presents the mathematical model for solving the aircraft scheduling

problem. Section 5 introduces the combined aircraft and crew scheduling problem and describes a heuristic that was implemented. An effective and more flexible method that can potentially be used in the future is introduced in Sect. 6. Results of computational experiments on each approach are provided. Finally, we conclude in Sect. 7 and propose directions for future work.

2 Aircraft and crew management

The services provided by FMCs are similar to those of a limousine service, but with guaranteed availability: To request a flight, a fractional owner calls the FMC and specifies the departure time, departure location and destination. There are multiple types of aircraft in the fleet differing in speed, seating capacity and operational cost. In many FMCs, an owner typically has shares in only one or two types of aircraft but can request any type for his/her trip (subject to provisional adjustments, e.g., in occupancy fees). At any point in time, the aircraft in the fleet may be at several different locations or airborne, and the ones ready for service may have to be repositioned from their current locations to the trip departure locations to serve the customers. Although the customer pays for the actual travel time, the FMC bears the repositioning cost. Depending on aircraft availability, schedulers may have to “downgrade” or “upgrade” some customers to different types of aircraft. Downgrades undermine owners’ goodwill while upgrades cost more for the FMC. Any trip request that cannot be met by company jets is satisfied by subcontracting to charter companies. Using a charter is significantly more expensive than using any of the company aircraft and the FMC bears the additional cost.

Each aircraft must go through maintenance at specified locations and time periods. In addition to maintenance, aircraft schedules must be coordinated with crew duty time constraints under the Federal Aviation Administration (FAA) regulations. Therefore managing the aircraft and crew is a multi-dimensional task for an FMC with the objectives of (i) minimizing empty aircraft moves and subcontracts, (ii) satisfying maintenance restrictions, (iii) meeting FAA regulations for the crew, and (iv) meeting customer requests on time. Note that the scheduling problem faced by the FMC is “hard”. Even without any consideration of the crew, scheduling the aircraft with maintenance restrictions is NP-complete (Keskinocak and Tayur 1998). In many FMCs, due to flight safety concerns and the operational convenience of rotating the crew, each aircraft and its crew are coupled together and regarded as a single entity for scheduling purposes (Martin et al. 2003). Hence, an aircraft is viewed as one operational unit which has to meet both maintenance restrictions and crew duty time limits. This approach is accepted by the FMC we worked with because of its consistency with the current practices in the industry and its simplicity in implementation.

Scheduling fractional ownership aircraft is first studied by Keskinocak and Tayur (1998) where a single type of aircraft is scheduled subject to maintenance and pre-scheduled trip restrictions. They report computational results for problems with up to 30 aircraft and 100 trips. Crew related issues are not considered. Ronen (2000) describes a decision support tool for scheduling charter aircraft considering crew availability and multiple types of aircraft. He proposes a set partitioning model to obtain optimal schedules, which is tested on problems with 24-hour, 36-hour and 48-hour long planning horizons, where 24–36 hours prove most effective. Computation times for at most 48 aircraft and 98 trips are reported. Martin et al. (2003) describe a decision support tool for fractional aircraft and crew scheduling where each aircraft and its crew are coupled. They model the problem as a 0-1 integer program (IP), which, incorporated into a decision support tool, provides exact solutions using ILOG

CPLEX. A recent paper by Hicks et al. (2005) discusses how Flexjet, a major FMC, improved its operations using a three module optimization system (with column generation) that simultaneously maximizes the use of aircraft, crew and facilities. No computation times are reported in the last two papers. Finally, Yao et al. (2004) propose a set partitioning formulation and column generation method for aircraft and crew scheduling. Their approach considers separation of crew from aircraft and presence of non-crew-compatible fleets. The effect of scheduled and unscheduled maintenance as well as flexible departure times are quantified by numerical experiments. Computation times for up-to 100 aircraft and 200 trips are reported.

During the development and implementation of our methods, research and development of similar tools for FMCs have proliferated as seen above. In this paper, we describe one particular scheduling tool by considering multiple types of aircraft, scheduled maintenance, and crew duty time constraints. Our models are tested on real world and randomly generated problem instances. We solve problems of significantly larger sizes than previously reported, mainly to address the current and anticipated future needs of the industry. We reported our preliminary results in Karaesmen et al. (2005) along with two 0-1 IP formulations for the scheduling problem: MIPS is similar to the formulations in Keskinocak and Tayur (1998) and Martin et al. (2003) and solves the aircraft scheduling problem. MIPS-CREW is developed for the combined aircraft and crew scheduling problem with 24-hour operations. MIPS and MIPS-CREW remain benchmarks for the methods we introduce in this paper.

3 Scheduling decision support tool

Traditionally, the daily schedules for the aircraft were manually prepared by experienced schedulers in the company. To meet the needs for future expansion and become more competitive in the industry, the FMC decided to automate scheduling decisions using an optimization based approach. Without an internal OR group, the company prefers a general purpose, commercial solver along with flexible, module based scheduling solutions that can be enhanced easily when appropriate. The FMC focuses on simplicity in development, implementation, and maintenance, and wants to stay away, to the extent possible, from customized heuristics.

FlightOptimizer, a decision support tool which is highly flexible for future enhancements, has been developed for managing aircraft scheduling, maintenance tracking, and flight release (Fig. 1). It can create a daily schedule from scratch, manually modify an existing schedule to incorporate new trips and flight changes, and evaluate the cost of a schedule and the utilization of the fleet. It consists of three modules: Optimizer Preparation Unit (OPU), Optimization Engine (OE) and Optimizer User Interface (OUI). Our team designed and developed OPU and OE. To create a feasible schedule, FlightOptimizer needs to check the feasibility of including a requested trip or a group of trips into a schedule. OPU serves this function (see Sect. 3.1). It retrieves the trip information, aircraft status and other related information from the central database server. It then processes the data and creates all the feasible trip assignments for an aircraft considering the positioning legs and required maintenance. Within OE there are two separate units: Model Creator (MC) and ILOG CPLEX solver. MC retrieves the information on feasible assignments from OPU and sets up the mathematical model for CPLEX. After a model is created, MC calls the routines from CPLEX, retrieves the output from the solver and passes the generated schedule to OUI. OUI presents the schedule via a user-friendly interface which shows each aircraft in the fleet and its assigned trips during the scheduling horizon. Schedulers have the flexibility of

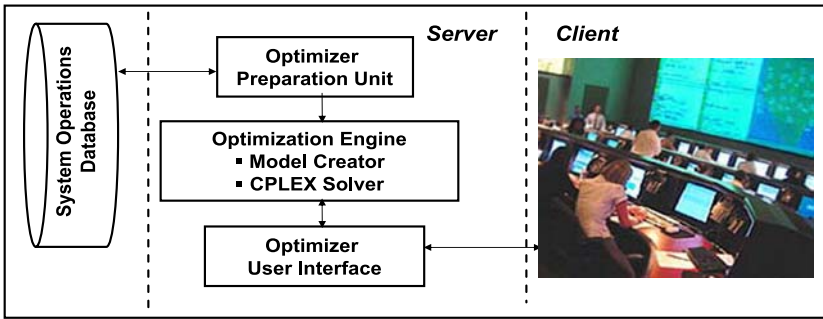


Fig. 1 Architecture of the scheduling decision support tool “FlightOptimizer”

trying different scenarios for what-if analysis or manually changing schedules in case of emergencies such as trip cancellation and aircraft breakdown.

A 24-hour planning horizon is adopted by the FMC due to the high uncertainty associated with flight requests outside this horizon. The practice is to create a *Master Schedule* (using OE) one day in advance since some airports need the flight and aircraft information at least 24 hours before the airport is used. The OUI, on the other hand, enables the schedulers to update the Master Schedule during the course of the horizon as new information becomes available.

3.1 Optimizer data preparation Unit (OPU)

The OPU is designed to preprocess the data to determine feasible aircraft-trip assignments. The earliest time an aircraft is ready for service, its initial location, and scheduled maintenance during the planning horizon are known a priori. For each trip we know (i) departure time, (ii) departure location, (iii) destination, (iv) type of aircraft requested by the customer, (v) aircraft that are incompatible with the trip, and (vi) whether or not the customer accepts chartered flights. Travel time between each pair of locations by each type of aircraft are also available.

We first preprocess the input data to identify the aircraft-trip compatibility and the availability of an aircraft to take a trip or a pair of trips back-to-back. Index i denotes aircraft and j represents trips. With a total of n aircraft of different types in the fleet and m requested trips, we store the preprocessed information in the matrices defined below.

$$AQ(i, j) = \begin{cases} 0, & \text{if aircraft } i \text{ is incompatible with trip } j \text{ given} \\ & \text{customer preferences;} \\ -1, & \text{if } i \text{ is compatible but is a downgrade for } j; \\ 1, & \text{if } i \text{ is compatible and equal in type or an upgrade for trip } j. \end{cases}$$

$$CQ(j) = \begin{cases} 1, & \text{if the customer requesting trip } j \text{ accepts a charter;} \\ 0, & \text{otherwise.} \end{cases}$$

$$AT(i, j) = \begin{cases} 1, & \text{if } AQ(i, j) \neq 0 \text{ and aircraft } i \text{ can serve the requested trip } j \text{ on time;} \\ 0, & \text{otherwise.} \end{cases}$$

$$ATT(i, j_1, j_2) = \begin{cases} 1, & \text{if aircraft } i \text{ can take trips } j_1 \text{ and } j_2 \text{ back-to-back;} \\ 0, & \text{otherwise.} \end{cases}$$

As mentioned before, each aircraft has to undergo pre-scheduled maintenance. The matrices AT and ATT are computed accordingly to ensure that an aircraft-trip assignment is deemed feasible only if the aircraft can complete the trip in the desired time frame without violating its scheduled maintenance. These matrices are passed to the OE to set up the mathematical model. Preprocessing dramatically reduces the number of decision variables by eliminating unacceptable/infeasible assignments. The computation time of these matrices is polynomial in n and m .

We aim to minimize the total cost in a 24-hour period while ensuring that each trip is served by either an aircraft in the fleet or a charter. Total cost consists of (i) repositioning costs, (ii) subcontracting costs and (iii) “hidden costs” which represent the cost of flying the aircraft while serving a trip and the upgrade/downgrade costs associated with that aircraft-trip assignment. We let $N = \{1, \dots, n\}$ and $M = \{1, \dots, m\}$ denote the set of all aircraft and trips, respectively. We use the following notation.

c_{i0j} = the cost of repositioning aircraft i from its initial location to the departure location of trip j , $i \in N$, $j \in M$.

$c_{ij_1j_2}$ = the cost of repositioning aircraft i from the destination of trip j_1 to the departure location of trip j_2 , $i \in N$, $j_1, j_2 \in M$.

b_j = the cost of subcontracting trip j , $j \in M$.

a_{ij} = the “hidden” cost while the customer occupies the aircraft, $i \in N$, $j \in M$.

For a trip (customer) that does not accept charters, i.e., $CQ(j) = 0$, we set $b_j = B_\infty$ where B_∞ denotes a high penalty cost. If $AQ(i, j) = 1$, then a_{ij} denotes the operational cost (of flying aircraft i while serving trip j). If $AQ(i, j) = -1$, then aircraft i is a downgrade for trip j . In this case, a_{ij} is higher than the operational cost since it includes a downgrade penalty, but it is not as high as b_j . There are no additional penalties for upgrades; FMC views them as complimentary. However, such upgrades are discouraged in this model by their hidden costs¹ whereas downgrades are penalized, but are preferred over charters.

3.2 Implementation and enhancements

We first developed a model, called NETIP, to solve the aircraft scheduling problem as requested by the FMC. NETIP is an integer program that is based on a minimum cost network flow formulation with side constraints. After NETIP was successfully tested within FlightOptimizer, the FMC showed an interest in extending the scope of the project to include crew related issues. Given the impending doubling of the fleet (due to a merger) and the urgency to automate scheduling decisions, NETIP was used in a heuristic—called RESTORE—to provide 24-hour schedules for combined aircraft and crew. It has been used by the company successfully since Spring 2002. A key performance indicator, measured carefully both by schedulers as well as senior management is *utilization*, defined as

$$\text{Utilization} = \frac{\text{Total flight time with passengers}}{\text{Total flight time with and without passengers}}.$$

¹An upgrade has a higher variable cost. Sheridan (2002) reports Citation V Ultra’s operating cost to be \$1,318, whereas a Boeing Business Jet costs \$4,360 per hour every time it leaves the hangar.

Since the implementation, the utilization has increased from 62% to over 70%.

In the remainder of the paper, we first introduce NETIP and compare it to a standard integer programming formulation—called MIPS—computationally. Although comparable to MIPS in terms of solution time, NETIP’s special structure makes it easy to build and incorporate into ILOG CPLEX. This gives a superior total setup and solution time to NETIP. Next, we briefly describe RESTORE. Finally, we propose a set partitioning model called BP (solved using branch-and-price) for the combined aircraft and crew scheduling problem. BP is flexible to represent various constraints regarding the maintenance and the crew (similar approaches are discussed in Hicks et al. 2005 and Yao et al. 2004). Since FlightOptimizer is designed to be a module-based decision support tool, alternative models such as BP can be easily incorporated into the OE to accommodate future operational requirements at the FMC. We present computational results on NETIP, MIPS, RESTORE, and BP based on both real-life and randomly generated data.

4 Aircraft scheduling

To determine the best aircraft-trip assignments in a given planning horizon, we developed NETIP, a minimum cost network flow model with side constraints. The model calls for creating a directed acyclic graph $G = (V, A)$ where V and A denote the set of nodes and arcs, respectively. The information from the OPU is used to construct G by defining the nodes and arcs (each with capacity 1) as follows:

Nodes: A source node s and a sink node t ; a node p_i for aircraft i , $i \in N$; auxiliary node u_{ij} for $i \in N$, $j \in M$; auxiliary node v_{ij} for $i \in N$, $j \in M$; a node r_j for trip j , $j \in M$.

Arcs: (s, p_i) and (s, u_{ij}) at cost 0, $i \in N$, $j \in M$; (r_j, t) at cost 0, $j \in M$; (p_i, v_{ij}) for $AT(i, j) = 1$ at cost c_{i0j} , $i \in N$, $j \in M$; (u_{ij_1}, v_{ij_2}) for $ATT(i, j_1, j_2) = 1$ at cost $c_{ij_1j_2}$, $i \in N$, $j_1, j_2 \in M$; (s, r_j) for trips j at cost b_j , $j \in M$; (v_{ij}, r_j) for $AT(i, j) = 1$ at cost a_{ij} .

An integer flow from node s to node t on this graph has the following interpretation: A positive flow on arc (v_{ij}, r_j) indicates trip j is assigned to aircraft i . A positive flow on arc (s, r_j) indicates trip j is subcontracted. A positive flow on (p_i, v_{ij}) indicates trip j is the first trip taken by aircraft i . A positive flow on arc (u_{ij_1}, v_{ij_2}) means that aircraft i takes trips j_1 and j_2 back-to-back. Arcs to/from node r_j ensure that trip j is taken by an aircraft in the fleet or is subcontracted.

The solution to the minimum cost network flow problem provides an integer flow on the graph, hence a trip-to-aircraft assignment. However, this assignment may not be the optimal solution of the aircraft scheduling problem unless side constraints are added. Consider the example with $n = 2$ and $m = 3$ in Fig. 2. Arcs on the graph represent feasible assignments and arc costs are given in Fig. 2(a). The optimal solution to the minimum cost network flow formulation assigns trips 1 and 2 to aircraft 1 and trip 3 to aircraft 1 (based on the flow on arcs (v_{ij}, r_j)). Note that this is not the optimal solution to the aircraft scheduling problem because the paths do not represent the correct sequence of trips for each aircraft and the cost of the network flow solution is not the actual cost of the schedule (see Fig. 2(b)). The optimal network flow solution indicates a positive flow on (u_{22}, v_{23}) with a repositioning cost for aircraft 2 from destination of trip 2 to departure location of trip 3. However, trip 2 is assigned to aircraft 1. This is contradictory. To eliminate such solutions from the network flow problem, we need to add side constraints: A trip j cannot precede another trip on aircraft i unless j is assigned to aircraft i (i.e., no flow allowed on arc (s, u_{ij}) unless there is positive flow on arc (v_{ij}, r_j)). Unfortunately, these side constraints destroy the total unimodularity of the constraint matrix, i.e., the linear programming (LP) relaxation of the model is not guaranteed to have an integer solution.

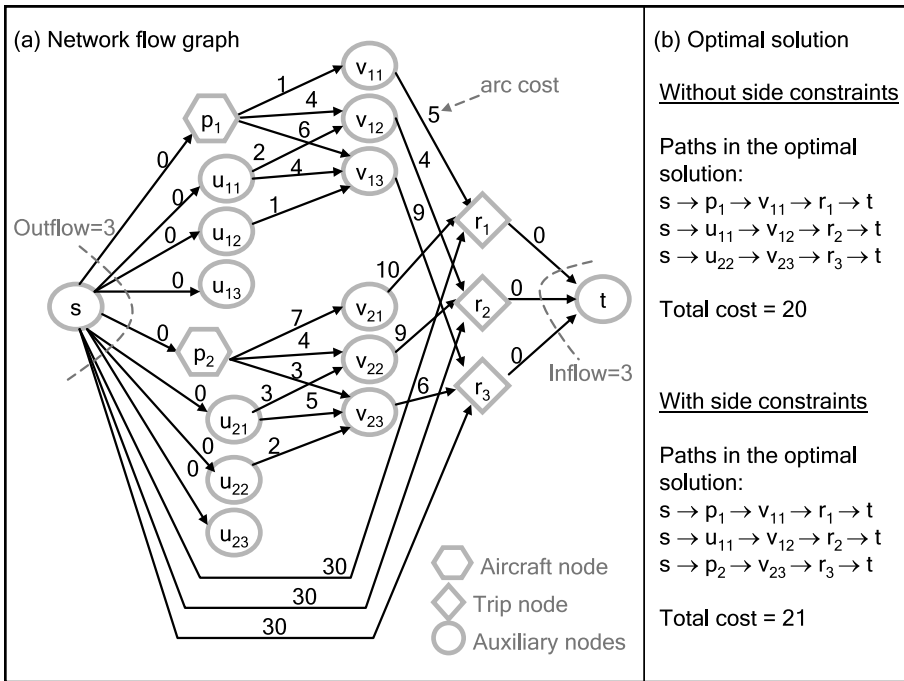


Fig. 2 An example of graph G and the minimum cost network flow based solutions

Let $x(e, f)$ to be the flow from node e to node f on arc (e, f) of graph G . The NETIP formulation is as follows:

$$\begin{aligned} \text{Min } & \sum_{i \in N} \sum_{\{j:AT(i,j)=1\}} c_{i0j} \cdot x(p_i, v_{ij}) \\ & + \sum_{i \in N} \sum_{\{j_1:AT(i,j_1)=1\}} \sum_{\{j_2:ATT(i,j_1,j_2)=1\}} c_{ij_1j_2} \cdot x(u_{ij_1}, v_{ij_2}) \\ & + \sum_{i \in N} \sum_{j \in M} a_{ij} \cdot x(v_{ij}, r_j) + \sum_{j \in M} b_j \cdot x(s, r_j) \end{aligned} \tag{1}$$

$$\text{s.t. } \sum_{i \in N} x(s, p_i) + \sum_{j \in M} x(s, r_j) + \sum_{i \in N} \sum_{j \in M} x(s, u_{ij}) = m, \tag{2}$$

$$\sum_{j \in M} x(r_j, t) = m, \tag{3}$$

$$x(s, p_i) - \sum_{\{j:AT(i,j)=1\}} x(p_i, v_{ij}) = 0, \quad i \in N, \tag{4}$$

$$x(s, u_{ij_1}) - \sum_{\{j_2:ATT(i,j_1,j_2)=1\}} x(u_{ij_1}, v_{ij_2}) = 0, \quad i \in N, \forall j_1 : AT(i, j_1) = 1, \tag{5}$$

$$x(s, r_j) + \sum_{i \in N} x(v_{ij}, r_j) - x(r_j, t) = 0, \quad j \in M, \tag{6}$$

Table 1 Performance comparison of NETIP and MIPS on real data

Instance (<i>n, m</i>)	Load factor (<i>m/n</i>)	Setup time		Solution time		Total time	
		NETIP	MIPS	NETIP	MIPS	NETIP	MIPS
(40, 41)	1.02	0.02	0.03	0.05	0.02	0.07	0.05
(44, 70)	1.59	0.19	0.58	1.16	0.42	1.35	1.00
(51, 35)	0.69	0.06	0.15	0.14	0.09	0.20	0.24
(55, 71)	1.29	0.09	0.19	0.27	0.12	0.36	0.31
(68, 135)	2.08	1.45	6.44	9.50	3.25	10.95	9.69
(72, 62)	0.86	0.15	0.41	0.41	0.24	0.56	0.65
(73, 90)	1.23	0.45	1.72	1.34	0.74	1.79	2.46
(74, 49)	0.66	0.10	0.24	0.31	0.15	0.41	0.39
(75, 105)	1.40	0.97	4.33	5.17	3.66	6.14	8.00
(76, 100)	1.32	0.46	1.44	2.24	0.71	2.70	2.15

$$x(p_i, v_{ij_2}) + \sum_{\{j_1: AT(i, j_1)=1\}} x(u_{ij_1}, v_{ij_2}) - x(v_{ij_2}, r_{j_2}) = 0, \quad j_2 \in M, \forall i : AT(i, j_2) = 1, \tag{7}$$

$$x(s, u_{ij}) - x(v_{ij}, r_j) \leq 0, \quad i \in N, \forall j : AT(i, j) = 1, \tag{8}$$

$$x(e, f) \in \{0, 1\}, \quad \forall (e, f) \in A. \tag{9}$$

The objective is to minimize the total cost. The first two terms in (1) are the repositioning costs whereas the last two terms represent the hidden costs of operation and the subcontracting cost, respectively. Constraints (2–7) ensure the flow balance of each node. (8) are the side constraints that prevent trip *j* to precede another trip on aircraft *i* unless *j* is assigned to aircraft *i*.

Let $\|AT\|$ and $\|ATT\|$ denote the number of non-zero coefficients in the AT and ATT matrices, respectively. NETIP has $(n + 2m + 3\|AT\| + \|ATT\|)$ variables and $(2 + n + m + 2\|AT\|)$ constraints. As an alternative to NETIP, we consider the IP formulation called MIPS of Karaesmen et al. (2005). MIPS has $(n + m + 2\|AT\| + \|ATT\|)$ variables and $(n + m + \|AT\|)$ constraints, significantly lower than NETIP.

We have tested NETIP and MIPS using both real world and randomly generated data sets. The results in Table 1 are based on real data from the winter season of 2001. The “Load Factor” is defined as *m/n*. “Setup Time” denotes the time spent to retrieve the preprocessed information and to prepare the formulations for CPLEX. “Solution Time” denotes the time for CPLEX to find an optimal integer solution. “Total Time” is the setup plus solution time. The problems are solved using ILOG CPLEX 7.5 on a PC with 1.13 GHz CPU and 512 MB memory, and all computation times are reported in CPU seconds. The setup time of NETIP is shorter than MIPS but its solution time is longer. The total computation time of NETIP (the sum of setup time and solution time) during this trial period was no more than 12 seconds, and in most cases less than 2 seconds, which is comparable to MIPS. These results show that the run time of OE is satisfactory when either NETIP or MIPS is used; however, this is not necessarily representative of the performance for future use. With the expected growth of the fleet, the daily scheduling problems at the FMC will be of significantly larger size. In addition, load factors can be as high as 2 to 3 during high seasons.

Table 2 Performance of NETIP and MIPS on randomly generated data over a 24-hour horizon

Experiment (<i>n</i> , <i>m</i>)	Load factor	Setup time		Solution time		Total time		B&B nodes	
		NETIP	MIPS	NETIP	MIPS	NETIP	MIPS	NETIP	MIPS
(50, 100)	2	0.05	0.32	0.53	0.64	0.58	0.96	0	0
(50, 150)	3	0.1	1.36	1.65	2.02	1.75	3.38	0	0
(75, 150)	2	0.16	3.41	3.89	4.53	4.05	7.94	0	0
(75, 225)	3	0.35	11.97	12.63	14.87	12.98	26.84	0	0
(100, 200)	2	0.39	14.16	13.7	14.98	14.09	29.14	0	0
(100, 300)	3	0.94	53.21	78.18	82.51	79.12	135.72	1	1
(125, 250)	2	0.87	49.05	54.18	60.68	55.05	109.73	1	1
(125, 375)	3	1.96	159.76	220.63	272.57	222.59	432.33	2	2
(150, 300)	2	1.48	109.43	133.44	122.56	134.92	231.99	0	1
(200, 400)	2	3.77	456.63	423.06	513.2	426.83	969.83	2	1

To examine the effect of fleet size and load factors on the computation time, we tested NETIP and MIPS using randomly generated data sets. The data is generated as follows: We first chose 100 cities that are uniformly distributed on a 100 by 100 grid. The flight time between two adjacent points on the grid is assumed to be 3 minutes, therefore the longest flight is about 7 hours. We randomly select (i) the initial location of the aircraft and (ii) a departure location and destination location for each trip with flight time greater than 30 minutes. The departure time for a trip is uniformly distributed between 0 and 900 minutes. The earliest available time of each aircraft is uniformly selected between 0 and 600 minutes. There are 8 types of aircraft in the fleet and the type of each aircraft is randomly assigned. Each trip requests a randomly chosen type of aircraft. All aircraft are assumed to be compatible with customer requests ($AQ(i, j) \neq 0$ for all i, j). During the planning horizon a randomly selected 20% of the fleet must go through a 150 minute-long maintenance. These data sets mimic the FMC's operations in a 24-hour period.

We use fleet sizes of 50–200 and load factors of 2–3 (yielding a total of 100–400 trips). We generate 10 independent, random data sets for each experiment. We report the average performance with respect to each experiment in Table 2. The solution time varies dramatically with fleet size and load factor. In majority of the experiments, NETIP's solution time is comparable to MIPS, and the LP relaxation of at least one of the formulations solves the problem optimally, i.e., zero branch-and-bound (B&B) nodes. However, overall NETIP has an advantage because of its lower total time. In summary, the computation times of NETIP and MIPS are comparable and negligible (no more than 2 CPU seconds) with smaller fleets and load factors. NETIP's total computation time is lower on average for moderate to high fleet sizes and load factors. However, there exist instances where MIPS is more efficient (this is also evident in our results reported in Karaesmen et al. 2005).

5 Combined aircraft and crew scheduling

We discussed earlier that FMCs prefer to couple aircraft and crew together, i.e., treat an aircraft and its assigned crew as a single resource. Under FAA regulations the duty period of a crew cannot exceed 14 hours (we call it *DUTY*) in a 24-hour period. Each duty period must be followed by a rest period of at least 10 consecutive hours (14–10 rule). When a crew

serves its first flight of the day (which can be a customer trip, or a repositioning flight), the duty period starts. Therefore, while creating a schedule, only those trips that can be served within *DUTY* hours after the duty period starts are eligible to be assigned to that aircraft. This restriction can be easily handled at the preprocessing stage if the duty start time of each crew is known. For instance, if an aircraft starts its duty at 9 a.m. and *DUTY* = 14 hours, the information in matrix *AT* can be updated easily to make sure that the aircraft does not take any trips that require flying past 11 p.m. Unfortunately, the only information that is available prior to creating a schedule is when the crew first becomes available for service. The actual duty start time of a crew-aircraft pair depends on the first trip assigned to that aircraft.

The crew-duty restriction can easily be incorporated into NETIP (or MIPS) by adding a constraint for each aircraft such that the time between the first take-off and the last landing of the aircraft is no more than *DUTY* (see the MIPS-CREW formulation in Karaesmen et al. 2005). However, once that constraint is added, the solution time of NETIP and MIPS increases significantly and obtaining an exact solution using a general purpose solver becomes impractical.

Due to the urgency in fully-automating and coordinating crew and aircraft decisions right after NETIP was developed, a heuristic called RESTORE was implemented at the FMC. The heuristic first modifies the *AT* and *ATT* matrices in the OPU to increase the likelihood of obtaining a crew-feasible solution, and then modifies this solution (if necessary) to ensure crew-feasibility. The heuristic assumes that there is a service period of \overline{DUTY} hours for each aircraft, which spans the period when the crew is “available”, i.e., it includes the duty time as well as the idle time before duty, when the crew is available but not used for service. Matrix *AT* is updated assuming that an aircraft will remain in service for the next \overline{DUTY} hours after it becomes available, and cannot take any trips that require flying beyond this duty period. Once an initial schedule is created based on preprocessing with a \overline{DUTY} value, RESTORE then modifies this schedule for feasibility as follows. First, it undoes all the aircraft-trip assignments that violate *DUTY* constraints. Next, it tries to assign each of these trips to the aircraft which can take the trip without exceeding *DUTY* at the lowest cost, while retaining its existing assignments. If no such aircraft exists, it subcontracts the trip. Choosing a conservative value for \overline{DUTY} , say $\overline{DUTY} = 14$ hours, will ensure feasibility of the initial schedule, but possibly lead to lower aircraft utilization and higher cost. On the other hand, longer \overline{DUTY} results in a more cost-efficient, but possibly infeasible initial schedule. We tested different \overline{DUTY} from 14 to 16 hours with increments of half an hour and found that $\overline{DUTY} = 14.5$ hours achieves the best tradeoff between schedule feasibility and efficiency in terms of total operational cost.

The computation time of RESTORE is determined by NETIP or MIPS because a crew-feasible schedule is obtained in polynomial time once the solution to the (adjusted) aircraft scheduling problem is given (see Karaesmen et al. 2005). The heuristic is not only useful for creating an aircraft and crew-feasible Master Schedule, but also practical for schedulers, who can update the Master Schedule throughout of the day manually. The OUI of FlightOptimizer enables schedulers to make necessary adjustments effectively.

6 Enhancements based on a set partitioning formulation

An alternative approach for solving the aircraft scheduling problem with crew duty time restrictions is to use a set partitioning formulation. The idea is to assign aircraft to feasible routes to ensure that each aircraft is assigned to at most one route and each trip is assigned

to an aircraft or is subcontracted. We use the following notation:

- Ω_i = the set of feasible routes for aircraft i , $i \in \bar{N}$,
- \bar{p} = a feasible route, $\bar{p} \in \Omega_i$,
- $\bar{c}_{i\bar{p}}$ = the total cost of taking route \bar{p} by aircraft i , $i \in \bar{N}$ and $\bar{p} \in \Omega_i$,
- $\bar{a}_{j\bar{p}}^i = 1$, if trip j is on route \bar{p} ; 0, otherwise, $\bar{p} \in \Omega_i$, $i \in \bar{N}$ and $j \in M$,
- $\theta_{i\bar{p}} = 1$, if aircraft i takes route \bar{p} ; 0, otherwise, $\bar{p} \in \Omega_i$ and $i \in \bar{N}$

where $\bar{N} = \{1, 2, \dots, n + m\}$ is the set of all aircraft such that $\{1, \dots, n\}$ represents the fleet and $\{n + 1, \dots, n + m\}$ represents charter aircraft for subcontracting the trips. For each $i \in \{n + 1, \dots, n + m\}$, Ω_i consists of a single route that takes only trip j with a cost of b_j .

The joint aircraft/crew scheduling problem is formulated as a set partitioning problem (SP):

$$(SP) \quad \min \sum_{i \in \bar{N}} \sum_{\bar{p} \in \Omega_i} \bar{c}_{i\bar{p}} \theta_{i\bar{p}} \tag{10}$$

$$\text{s.t.} \quad \sum_{i \in \bar{N}} \sum_{\bar{p} \in \Omega_i} \bar{a}_{j\bar{p}}^i \theta_{i\bar{p}} = 1, \quad j \in M, \tag{11}$$

$$\sum_{\bar{p} \in \Omega_i} \theta_{i\bar{p}} \leq 1, \quad i \in \bar{N}, \tag{12}$$

$$\theta_{i\bar{p}} \in \{0, 1\}, \quad i \in \bar{N}, \text{ and } \bar{p} \in \Omega_i. \tag{13}$$

Constraints (11) ensure that each trip is covered and constraints (12) ensure that an aircraft is assigned to at most one route. To obtain an optimal integral solution, we use the branch and price method (BP) which combines column generation with branch-and-bound (see Lubbecke and Desrosiers 2005; Savelsbergh 1997; Barnhart et al. 1998).

6.1 Column generation

We call the linear relaxation of SP the master problem (MP). In column generation we first solve a restricted master problem (RMP), which has the same objective function and constraints (11) and (12) as MP except that a subset of columns $\Omega' \subseteq \Omega$ is used, where $\Omega'_i \subseteq \Omega_i$, $\Omega' = \Omega'_1 \cup \Omega'_2 \cup \dots \cup \Omega'_{n+m}$, and $\Omega = \Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_{n+m}$. Constraints (13) are replaced by non-negativity constraints in RMP.

In RMP, dual variables α_j and γ_i associated with constraints (11) and (12), respectively, are then created and passed to the following pricing subproblem (PP) to price out favorable columns, i.e., to generate new columns which are added to the RMP.

$$(PP) \quad \arg \min \left\{ \tilde{c}_{i\bar{p}} = \bar{c}_{i\bar{p}} - \sum_j \alpha_j \bar{a}_{j\bar{p}}^i - \gamma_i \mid i \in \bar{N}, \bar{p} \in \Omega_i \right\} \tag{14}$$

In our application, the initial basis for solving RMP can be chosen such that each trip is subcontracted. This choice is trivial and an alternative (better) initial basis may be created, e.g., using a heuristic. We have experimented with several different alternatives and

- Given** $i \in N$: Initialize $d(j) = \infty$ for $j \in M$.
- Step 1:* Set $T^i = \{j \mid j \in M, AT(i, j) = 1, j \text{ crew-feasible for } i\}$.
- Step 2:* For $j \in T^i$, let j be first trip on any path, i.e.
 $d(j) = c_{i0j} + a_{ij} - \alpha_j - \gamma_i$ and $pred(j) = 0$.
- Step 3:* Set $T = T^i$. Create multi-trip paths:
Step 3.1: (Node selection)
 If $T = \emptyset$, goto Step 4. Otherwise,
 let $j_1 = \arg \min_j \{d(j) \mid j \in T\}$; set $T = T - \{j_1\}$.
- Step 3.2:* (Distance update)
 For each $j_2 \in T^i$ where $ATT(i, j_1, j_2) = 1$
 and crew duty constraint satisfied, do
 If $d(j_2) > d(j_1) + c_{ij_1j_2} + a_{ij_2} - \alpha_{j_2}$,
 then $d(j_2) = d(j_1) + c_{ij_1j_2} + a_{ij_2} - \alpha_{j_2}$ and $pred(j_2) = j_1$.
 Goto Step 3.1.
- Step 4:* If $\min\{d(j) \mid j \in T^i\} < 0$, then determine $j^* = \arg \min_j \{d(j) \mid j \in T^i\}$
 (i.e. the path to trip j^* has the most-negative reduced cost).

Fig. 3 Algorithm to find most-negative reduced cost route for aircraft i

observed that starting with the trivial solution is a fairly good choice for our application. If it does not conform to the structure of the optimal solution, even a good initial solution is detrimental to the pricing step. The above finding is also consistent with the experience shared in Vanderbeck (1994).

Once the RMP is solved with the current set of columns, we then look for a new column (route) that can be added to the RMP, with the most negative reduced cost. Computing the reduced cost of subcontracting the trips is trivial. For the aircraft in the fleet, we solve a constrained shortest path problem for each $i \in N$ to determine a candidate route to be added to RMP. We utilize the pre-processed matrices such as AT and ATT to ensure the feasibility of a route for both the aircraft and the crew rather than imposing explicit constraints in the pricing problem. For longer than 24 hours, where a crew may have multiple duty and rest periods, a new set of matrices similar to AT and ATT can be created (see Martin et al. 2003) to identify feasible routes. The algorithm provided in Fig. 3 guarantees a feasible (if not optimal) solution to the pricing problem and is very efficient. It is a modified version of Dijkstra’s shortest path algorithm where we seek shortest paths from the source node (each aircraft i) to the trip nodes $j \in M$; $d(j)$ denotes the shortest distance from the source node to trip node j , and $pred(j)$ denotes the predecessor of j .

6.2 Branch and price

When the solution of the RMP is not integral, we choose a column with a fractional solution to branch. Our proposed branching rule uses similar ideas as in Ryan and Foster (1981), adapted for multiple types of aircraft.

Branching rule: Consider two trips $j, j' \in M$ and a basic route \bar{p} of the Master Problem where $\theta_{i\bar{p}}$ is fractional. Let trip j be any trip with $\bar{a}_{j\bar{p}}^i = 1$. Since $\theta_{i\bar{p}}$ is fractional and $\sum_{i \in \bar{N}} \sum_{\bar{p} \in \Omega_i} \bar{a}_{j\bar{p}}^i \theta_{i\bar{p}} = 1$, there must exist some other basic route \bar{p}' with $0 < \theta_{i'\bar{p}'} < 1$ and $\bar{a}_{j'\bar{p}'}^{i'} = 1$, where $i' \in \bar{N}$. Because there are no duplicate columns in the basis, there are two cases to consider: First, if $i' = i$, then there must exist another trip j' such that either

$\bar{a}^i_{j'\bar{p}'} = 1$ or $\bar{a}^i_{j\bar{p}} = 1$ but not both, which leads to

$$1 = \sum_{i \in \bar{N}} \sum_{\bar{p} \in \Omega_i} \bar{a}^i_{j\bar{p}} \theta_{i\bar{p}} = \sum_{\{\bar{p} \in \Omega_i: \bar{a}^i_{j\bar{p}}=1\}} \theta_{i\bar{p}} > \sum_{\{\bar{p} \in \Omega_i: \bar{a}^i_{j\bar{p}}=\bar{a}^i_{j'\bar{p}'}=1\}} \theta_{i\bar{p}} > 0. \tag{15}$$

Therefore, we have

$$0 < \sum_{\{\bar{p} \in \Omega_i: \bar{a}^i_{j\bar{p}}=\bar{a}^i_{j'\bar{p}'}=1\}} \theta_{i\bar{p}} < 1 \quad \text{for } i \in \bar{N}. \tag{16}$$

Second, if $i' \neq i$, then trip j cannot be the only trip on both \bar{p} and \bar{p}' ; otherwise integral solutions must be obtained for $\theta_{i\bar{p}}$ and $\theta_{i'\bar{p}'}$. Therefore, there must exist another trip j' such that either $\bar{a}^i_{j'\bar{p}} = 1$ or $\bar{a}^i_{j'\bar{p}'} = 1$ or both. Such two trips j and j' give a pair of branching constraints

$$\sum_{\{\bar{p} \in \Omega_i: \bar{a}^i_{j\bar{p}}=\bar{a}^i_{j'\bar{p}}=1\}} \theta_{i\bar{p}} = 1, \quad \sum_{\{\bar{p} \in \Omega_i: \bar{a}^i_{j\bar{p}}=\bar{a}^i_{j'\bar{p}'}=1\}} \theta_{i\bar{p}} = 0 \quad \text{for } i \in \bar{N}. \tag{17}$$

Constraints in (17) are analogous to requiring trips j, j' and aircraft i be bundled together and covered by the same column on the first (left) branch, and be disjoint and covered by different columns on the second (right) branch. Columns in the RMP violating the branching constraints are removed and only the ones satisfying the branching constraints are included in the pricing problem. Since for each $i \in \bar{N}$ the number of branching pairs is finite, the algorithm terminates after a finite number of branches, with an integral solution.

A common approach for selecting a node for branching is to use depth first search. Using (16) and (17), we first explore the left branch of a node since the subproblem on the left branch is easier to solve in our application than the one on the right branch. We can identify a subset of trips $M' \subseteq M$ with $|M'| \geq 2$ such that

$$0 < \sum_{\{\bar{p} \in \Omega_i: \bar{a}^i_{j\bar{p}}=1, \forall j \in M'\}} \theta_{i\bar{p}} < 1, \quad \text{for } i \in \bar{N}. \tag{18}$$

The presence of multiple-types of aircraft increases the number of feasible columns dramatically. Rather than selecting a pair of trips, we can bundle all trips in M' with aircraft i together on the left branch and disjoin them on the right. Since this imposes tighter constraints on the left branch, this branching rule is especially helpful in reaching a feasible solution quickly.

6.3 Computational performance of BP

We test BP and RESTORE on the same random data sets used in Sect. 4 with 24-hour scheduling horizons. Figure 4 reports the average and maximum percentage optimality gaps of RESTORE and BP. In these experiments, the optimal solution is obtained by MIPS-CREW introduced in Karaesmen et al. (2005). The computation times of BP and MIPS-CREW are presented in Table 3. Note MIPS-CREW is too slow for large problems; we do not report the results of experiments for which MIPS-CREW took more than 5 hours for at least one instance. Computation time of RESTORE is similar to that of NETIP and is not reported here. In Table 3, we also report the computation time of BP for 48-hour horizon problems for which the random data is created in a similar way to the experiments

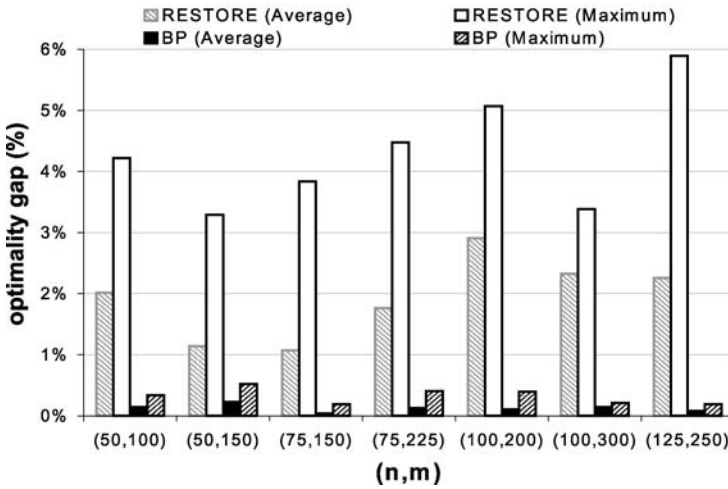


Fig. 4 Optimality gap of BP and RESTORE in various experiments

Table 3 Performance of BP and MIPS-CREW on randomly generated data

Experiment (n, m)	Load factor	24-hour horizon				48-hour horizon	
		BP		MIPS-CREW		BP	
		Solution time	Utilization	Solution time	Utilization	Solution time	Utilization
(50, 100)	2	0.49	0.771	25.63	0.768	1.3	0.77
(50, 150)	3	0.80	0.791	367.8	0.782	4.34	0.762
(75, 150)	2	1.45	0.79	142.16	0.785	6.74	0.787
(75, 225)	3	2.67	0.808	3862.22	0.799	45.96	0.788
(100, 200)	2	3.83	0.807	498.47	0.807	18.38	0.803
(100, 300)	3	7.69	0.82	23857.63	0.82	165.52	0.804
(125, 250)	2	9.53	0.821	1780.33	0.818	60.95	0.818
(125, 375)	3	18.37	0.829	>5 hours	–	310.31	0.818
(150, 300)	2	17.66	0.829	>5 hours	–	164.33	0.825
(200, 400)	2	54.72	0.844	>5 hours	–	606.46	0.823

in Section 4 except that the departure time of a trip is uniformly distributed between 0 and 1800 minutes and the earliest available time of each aircraft is uniformly selected between 0 and 1200 minutes. BP solves the 48-hour problem in around 10 minutes on average even for large size problems; this is within the acceptable limit of 15 minutes set by the FMC.

In general, RESTORE is a promising alternative for practical purposes in terms of simplicity, solution quality and computation time for small size problems in a 24-hour scheduling horizon. However, for large size problems BP is more robust and efficient. The real benefit of BP is its ability to solve scheduling problems with added features such as longer horizons.

Table 4 Performance of BP on real data from two different FMCs

Experiment (<i>n</i> , <i>m</i>)	Scheduling horizon (days)	Solution time	Subcontracted Trips	Utilization
(37, 53)	2	2.93	0	0.84
(37, 81)	3	9.22	1	0.83
(37, 113)	4	40.71	1	0.85
(37, 177)	5	3932.05	6	0.81
(37, 207)	7	6575.31	12	0.80
(211, 312)	2	257.27	7	0.91

To show the strength of BP, we also use a set of real data provided by two different FMCs. The first five instances in Table 4 are from one FMC with scheduling horizons of 2–7 days. The last instance is from a second FMC in October 2004. Notice that the solution time increases significantly beyond 4 days even for small fleet sizes. Despite the high uncertainty associated with trip requests and aircraft availability, 2- or 3-day rolling horizon schedules will give the FMC an advantage by potentially decreasing repositioning times at the beginning of each day. Our implementation of BP provides a satisfactory run-time for such problems.

7 Conclusions and future research

We described a scheduling decision support tool, FlightOptimizer, for a major fractional management company (FMC) that operates fractional ownership aircraft. We first implemented a network flow model, called NETIP, for the aircraft scheduling problem. Because of its simplicity in implementation, NETIP has been used in conjunction with RESTORE, which creates crew-feasible schedules, since Spring 2002. The average utilization of created schedules—a performance indicator used by senior executives—has increased to over 70%, compared to 62% before the implementation. In addition to RESTORE, we present an alternative method, BP, for solving the combined aircraft and crew scheduling problem of FMCs.

Our computational results indicate that NETIP is a good choice for small to medium fleet sizes and load factors. RESTORE can generate crew-feasible schedules quickly with an average optimality gap of 3% for a 24-hour planning horizon. However, to solve large size problems covering longer planning horizons and to incorporate complicated operational rules, BP is a better alternative, resulting in near-optimal solutions with less than 0.5% optimality gap on average. With our flexible, module-based decision support tool, alternative optimization models such as BP can be easily incorporated into FlightOptimizer to accommodate different operational requirements.

Two extensions to our models deserve investigation: First, in fractional ownership programs both demand and aircraft supply change dynamically over time. For instance, for the FMC we worked with, around 25% of the requested trips arrive after the original schedule is created, and each day 10–12 aircraft require unplanned maintenance. Current schedule optimization is done by assuming that the demand and supply information is deterministic over the planning horizon. When there is a change on either side, the schedule has to be modified, but has to remain close to the original one because of several practical factors. A common complaint from managers when using optimization-based decision support

tools is that small changes in input often translate into dramatically different solutions that are sometimes managerially unacceptable (Brown et al. 1997). Therefore, enhancements to FlightOptimizer so that it can update the schedule to accommodate disruptions in real time while keeping it as close to the original one as possible are desirable. Second, current scheduling solutions consider all the trip requests as firm demand that must be satisfied by company aircraft or charter aircraft. However, it is possible that some demand is optional, i.e., should be satisfied only if it is profitable. The FMC currently sees at least two sources of optional demand: Charter requests from other companies and special requests from customers in excess of their available flight hours. We believe enhancing the current models to incorporate optional demands is an interesting and challenging extension.

Acknowledgements We would like to thank Brian Koluder, Ljuba Masharina, Patrick Golembiewski, and Hank Posta for their cooperation during the development of the decision support tool. Pinar Keskinocak was supported by NSF Career Award DMI-0093844.

References

- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., & Vance, P. H. (1998). Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46(3), 316–329.
- Brown, G., Dell, R., & Wood, R. (1997). Optimization and persistence. *Interfaces*, 27(5), 15–37.
- Hicks, R., Madrid, R., Milligan, C., Pruneau, R., Kanaley, M., Dumas, Y., Lacroix, B., Desrosiers, J., & Soumis, F. (2005). Bombardier flexjet significantly improves its fractional aircraft ownership operations. *Interfaces*, 35(1), 49–60.
- Karaesmen, I., Keskinocak, P., Tayur, S., & Yang, W. (2005). Scheduling time-shared aircraft: models and methods for practice. In *Proceedings of MISTA 2005: the 2nd multidisciplinary conference on scheduling: theory & applications* (Vol. 1, pp. 19–38), New York, 18–21 July 2005.
- Keskinocak, P. (1999). Corporate high flyers. In *OR/MS today* (pp. 22–25), December 1999.
- Keskinocak, P., & Tayur, S. (1998). Scheduling of time-shared jet aircraft. *Transportation Science*, 32(3), 277–294.
- Lubbecke, M. E., & Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6), 1007–1023.
- Martin, C., Jones, D., & Keskinocak, P. (2003). Optimizing on-demand aircraft schedules for fractional aircraft operators. *Interfaces*, 33(3), 22–35.
- NBAA. (2004). *National business aviation association fact book*. Washington, DC. Available at <http://web.nbaa.org/public/news/stats/factbook/2004/2004factbook.pdf>.
- Ronen, D. (2000). Scheduling charter aircraft. *Journal of the Operational Research Society*, 51, 258–262.
- Ryan, D. M., & Foster, B. A. (1981). An integer programming approach to scheduling. In A. Wren (Ed.), *Computer scheduling of public transport urban passenger vehicle and crew scheduling* (pp. 269–280). Amsterdam: North-Holland.
- Savelsbergh, M. W. P. (1997). A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, 45(6), 831–841.
- Sheridan, P. (2002). Some advantages of private aviation, In *Pittsburgh post gazette*, 21 January 2002, p. C-3. <http://www.post-gazette.com/seen/stylefile/20020121style0121p1.asp>.
- Sweetman, B. (2005). Flight to convenience. *Fortune*, 152(8), S1–S7.
- Vanderbeck, F. (1994). *Decomposition and column generation for integer programs* (Technical report). Belgium: Universite Catholique de Louvain.
- Yao, Y., Ergun, O., Johnson, E., Schultz, W., & Singleton, J. M. (2004). *Strategic planning problems in fractional aircraft ownership programs*. Working Paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.