



INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Integrated Airline Schedule Design and Fleet Assignment: Polyhedral Analysis and Benders' Decomposition Approach

Hanif D. Sherali, Ki-Hwan Bae, Mohamed Haouari,

To cite this article:

Hanif D. Sherali, Ki-Hwan Bae, Mohamed Haouari, (2010) Integrated Airline Schedule Design and Fleet Assignment: Polyhedral Analysis and Benders' Decomposition Approach. INFORMS Journal on Computing 22(4):500-513. <http://dx.doi.org/10.1287/ijoc.1090.0368>

Full terms and conditions of use: <http://pubsonline.informs.org/page/terms-and-conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2010, INFORMS

Please scroll down for article—it is on subsequent pages



INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Integrated Airline Schedule Design and Fleet Assignment: Polyhedral Analysis and Benders' Decomposition Approach

Hanif D. Sherali, Ki-Hwan Bae

Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061 {hanifs@vt.edu, kbae04@vt.edu}

Mohamed Haouari

Department of Industrial Engineering, Ozyegin University, 34662 Istanbul, Turkey; and King Saud University, Riyadh 11421, Saudi Arabia, mohamed.haouari@ozyegin.edu.tr

The main airline operations consist of schedule planning, fleet assignment, aircraft routing, and crew scheduling. To improve profitability, we present in this paper an integrated fleet assignment model with schedule planning by simultaneously considering optional flight legs to select along with the assignment of aircraft types to all scheduled legs. In addition, we consider itinerary-based demands for multiple fare classes. A polyhedral analysis is conducted of the proposed mixed-integer programming model to tighten its representation via several classes of valid inequalities. Solution approaches are developed by applying Benders' decomposition method to the resulting lifted model, and computational results are presented using real data obtained from a major U.S. airline to demonstrate the efficacy of the proposed procedures.

Key words: integrated airline operations; flight scheduling; fleet assignment; valid inequalities; Benders' decomposition

History: Accepted by Karen Aardal, Area Editor for Design and Analysis of Algorithms; received January 2009; revised July 2009; accepted September 2009. Published online in *Articles in Advance* December 30, 2009.

1. Introduction

As airlines cut capacities or pursue mergers and alliances as a result of high rising fuel costs and weak demands, the trend is to reduce operations on routes that are not profitable, retire older and fuel-inefficient aircraft, and increase fares. Considering that an increase in load factors is desirable to achieve a balance between demand and supply, airlines need to generate more profitable schedules and to assign aircraft having sufficient, but not excessive, capacity to accommodate demands based on aircraft availabilities, operational costs, and potential revenues. These aspects, respectively, relate to the schedule planning and fleet assignment problems.

Abara (1989) limited the number of possible feasible connections for each flight leg to facilitate the fleet assignment process, whereas Subramanian et al. (1994) implemented the Coldstart model to address the assignment of aircraft to legs at Delta Airlines. A fleet assignment model that is still popularly used was formulated by Hane et al. (1995), who assumed a repeated daily schedule and proposed several preprocessing measures such as node aggregation and the identification of isolated islands at stations, all of which have proven to be critical in reducing

the network size and thus the problem complexity. Clarke et al. (1996) generalized the basic fleet assignment model by including maintenance and crew scheduling considerations while preserving its solvability. Rushmeier and Kontogiorgis (1997) modeled the fleet assignment problem as a mixed-integer multicommodity flow problem, where the commodities are fleets and the constraints satisfy the different operational requirements and ensure the coverage of flights. Kniker (1998), Jacobs et al. (1999), and Barnhart et al. (2002) considered origin-destination (OD) fleet assignment models, where passenger revenues are accounted for each OD itinerary by using a passenger-mix model to ascertain booking levels for each fixed-seat capacity aircraft.

As fleet assignment decisions need to be made well in advance of departures for the purpose of scheduling crews (although demand is highly uncertain at this point), Sherali et al. (2005) proposed a mixed-integer programming demand-driven reflighting model to be solved subsequently under more accurate demand forecasts while considering path-level or itinerary-level demands as in Barnhart et al. (2002), where the reassignments of aircraft to flight legs are limited within the same family of aircraft so as to preserve serviceability by the same

crew. The advantage of this approach lies in the potential utilization of a more accurate and detailed demand forecast as departure times approach. Furthermore, Sherali and Zhu (2008) introduced a two-stage stochastic programming approach where the first stage performs the initial fleet assignment while recognizing that the subsequent re-fleeting process can make adjustments in the second stage in response to the stochastic demand fluctuations. Pilla et al. (2008) employed a similar two-stage stochastic programming framework and used a statistical experimental approach along with response surface techniques to approximate the optimal expected profit function. Sherali et al. (2006) provided a recent and comprehensive survey on airline fleet assignment models.

Integrated fleet assignment and flight scheduling considerations can increase revenues by allowing for improved flight connection opportunities in the fleet assignment problem. Desaulniers et al. (1997) first addressed this issue by permitting the departure times of legs to vary within specified time-windows, where the resulting model was solved using a branch-and-price approach. If a tentative flight schedule is at hand that includes certain optional legs, then the decision regarding which optional legs to offer can be made concurrently with the fleet assignment model. However, changes in the schedule could possibly affect demands. For example, the deletion of a leg may increase the demands on paths having origins, destinations, and time frames that are compatible with the paths that contain the particular leg. Such considerations were accommodated by Lohatepanont and Barnhart (2004) in their integrated schedule design and fleet assignment model, which also simultaneously determines how many passengers to spill on each path using itinerary-based demands and what proportion of these spilled passengers to recapture on other compatible itineraries.

The ever-expanding literature on airline operational problems variously demonstrates the benefits of examining integrated models and, in particular, applying Benders' decomposition method to solve such large-scale problems. Jacobs et al. (1999) proposed an OD fleet assignment problem and solved it using Benders' decomposition while incorporating passenger flow considerations. In generating schedules from scratch, Lettovsky et al. (1999) used Benders' methodology to solve a model for coordinating the travel itineraries of multiple travelers coming from multiple origin locations. Benders' decomposition has also been used to solve an integrated aircraft routing and crew scheduling problem by Cordeau et al. (2001), Mercier et al. (2005), and Mercier and Soumis (2007), whereas Li et al. (2006) used it for a simultaneous fleet assignment and cargo routing problem. Sandhu and Klabjan

(2007) integrated all airline scheduling stages and proposed two solution approaches: Benders' decomposition and a combination of Lagrangian relaxation and column generation. Alternative Benders' decomposition and branch-and-price algorithms were designed by Haouari et al. (2009) to solve an integrated aircraft fleet and routing problem. More recently, Papadakos (2009) presented several integrated models for airline scheduling, which were solved by applying a Benders' decomposition method combined with column generation.

In this paper, we develop and analyze a model that integrates flight scheduling and fleet assignment, while considering optional legs, path- and itinerary-based demands, and multiple fare classes. In the model presented by Lohatepanont and Barnhart (2004), the different demand parameters and the demand correction terms are estimated and revised iteratively using a schedule evaluation package that takes the resulting flight schedule as an input. This requires several feedback loops between the model and the evaluation package. In addition, they treat the passengers accepted on different itineraries in a separate passenger-mix model that is solved subsequently, using the optional leg selections and fleet assignment decisions as given by the main model. In contrast, we integrate this latter feature within our main model itself. Furthermore, we perform a polyhedral analysis to tighten the model representation and propose an alternative specialized solution approach based on Benders' decomposition and a sequential fixing process. However, because our model is intended to serve as an initial stand-alone evaluation tool, we assume that the leg selections do not affect path demands, and we also tentatively ignore the recapture effect. In this case, instead of defining the model in terms of the extent of spillage, it is more appropriate to define variables that directly represent the accepted demand for each itinerary. Having ascertained the final schedule based on the prescribed selection of optional legs, we can assess the effect this has on demand using an appropriate schedule evaluation package as in Lohatepanont and Barnhart (2004) and reiterate the solution process as necessary.

Our paper makes the following specific contributions. First, we present a new mixed-integer programming model that integrates flight scheduling and fleet assignment, and directly incorporates multiple fare classes and the number of passengers to accept on each active path. Second, certain classes of valid inequalities are introduced through a polyhedral analysis and are either a priori accommodated within the model or are successively generated via suitable separation problems to tighten the model representation and reduce its complexity. Third, we

propose a novel Benders' decomposition-based solution approach that is suitable for handling large-scale problems. Finally, we present computational results using real data provided by a major U.S. airline carrier. Our computational study demonstrates the efficacy of the proposed procedures and reveals a substantial potential increase in profitability.

The remainder of this paper is organized as follows. In §2, we present the basic integrated mixed-integer programming model. Section 3 introduces several classes of valid inequalities along with accompanying separation routines to generate them for tightening the model representation. In §4, we discuss two different algorithmic approaches using Benders' decomposition to efficiently optimize the developed model. Results from our computational experiments using real airline data are reported and analyzed in §5, and finally, §6 provides a summary of the paper along with some concluding remarks.

2. Model Description and Notation

Prior to describing our model, we present the following notation that is based on a standard time-space network representation for all the fleet types (for example, see Berge and Hopperstad 1993, Hane et al. 1995, Sherali et al. 2005).

Sets

AT: set of aircraft types, indexed by a .

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

$L^M \subseteq L$: set of mandatory legs, indexed by j .

$L^O \subseteq L$: set of optional legs that are candidates for deletion, indexed by j .

N_a : set of nodes in aircraft type a 's network, $a \in AT$; indexed by n .

G_a : set of ground arcs in aircraft type a 's network, $a \in AT$; indexed by g .

CS_a : set of arcs passing forward in time through a counting time line in aircraft type a 's network, $a \in AT$.

Π : set of all paths (related to considered itineraries), indexed by p .

$\Pi^O \subseteq \Pi$: set of paths containing any of the optional legs (hence, subject to deletion), indexed by p .

$\Pi(j)$: set of paths in Π that contain leg j , $\forall j \in L$ ($\Pi^O(j)$ is defined similarly).

$L(p)$: set of legs belonging to path p , $\forall p \in \Pi$.

$L^O(p)$: set of optional legs belonging to path p , $\forall p \in \Pi^O$ ($L^O(p) = L(p) \cap L^O$).

H : set of all fare classes, indexed by h .

$H_p \subseteq H$: set of all fare classes on path p , $\forall p \in \Pi$; indexed by h .

Parameters

c_{aj} : cost of assigning fleet type a to leg j , $\forall a \in AT$, $j \in L$.

NA_a : number of available aircraft for fleet type a , $\forall a \in AT$.

Cap_{ah} : capacity of aircraft type a to accommodate passengers for fare class h , $\forall a \in AT$, $h \in H$.

μ_{ph} : mean demand for fare class h on path (or itinerary) p , $\forall p \in \Pi$, $h \in H_p$.

f_{ph} : estimated price for fare class h on path p , $\forall p \in \Pi$, $h \in H_p$.

$$bf_{jn} \begin{cases} 1, & \text{if flight } j \text{ begins at node } n \\ & \text{(in aircraft type } a \text{'s network);} \\ -1, & \text{if flight } j \text{ ends at node } n \\ & \text{(in aircraft type } a \text{'s network);} \\ 0, & \text{otherwise, } \forall j \in L, n \in N_a, a \in AT. \end{cases}$$

$$bg_{gn} \begin{cases} 1, & \text{if ground arc } g \text{ begins at node } n \\ & \text{(in aircraft type } a \text{'s network);} \\ -1, & \text{if ground arc } g \text{ ends at node } n \\ & \text{(in aircraft type } a \text{'s network);} \\ 0, & \text{otherwise, } \forall g \in G_a, n \in N_a, a \in AT. \end{cases}$$

Decision Variables

$$x_{aj} \begin{cases} 1, & \text{if fleet type } a \text{ covers leg } j; \\ 0, & \text{otherwise, } \forall a \in AT, j \in L. \end{cases}$$

w_g : number of aircraft (of type a) on ground arc g in aircraft type a 's network, $\forall g \in G_a$, $a \in AT$.

$$z_p \begin{cases} 1, & \text{if path } p \text{ is included in the flight network;} \\ 0, & \text{otherwise, } \forall p \in \Pi^O. \end{cases}$$

π_{ph} : number of passengers in fare class h accepted on path p , $p \in \Pi$, $h \in H_p$ (undefined π -values are assumed to be zero).

Accordingly, we formulate an initial, simple integrated *flight scheduling, and fleet assignment model* (FSFAM1) as follows:

$$(FSFAM1) \quad \text{Maximize } \sum_{p \in \Pi} \sum_{h \in H_p} f_{ph} \pi_{ph} - \sum_{a \in AT} \sum_{j \in L} c_{aj} x_{aj} \quad (1)$$

subject to

$$\sum_{a \in AT} x_{aj} = 1, \quad \forall j \in L^M, \quad (2)$$

$$\sum_{a \in AT} x_{aj} \leq 1, \quad \forall j \in L^O, \quad (3)$$

$$\sum_{j \in L} bf_{jn} x_{aj} + \sum_{g \in G_a} bg_{gn} w_g = 0, \quad \forall n \in N_a, \forall a \in AT, \quad (4)$$

$$\sum_{j \in CS_a} x_{aj} + \sum_{g \in CS_a} w_g \leq NA_a, \quad \forall a \in AT, \quad (5)$$

$$\sum_{p \in \Pi(j)} \pi_{ph} \leq \sum_{a \in AT} Cap_{ah} x_{aj}, \quad \forall j \in L, \forall h \in H, \quad (6)$$

$$\pi_{ph} \leq \mu_{ph}, \quad \forall p \in \Pi, \forall h \in H_p, \quad (7)$$

$$x: \text{ binary}, \quad (w, \pi) \geq 0. \quad (8)$$

The objective function (1) is to maximize the net revenue considering multiple fare classes for each path or itinerary. The cover constraints (2) and (3) distinguish between the set of mandatory and optional legs. Constraints (4) and (5) are, respectively, the conservation of flow and the aircraft resource count restrictions. Constraint (6) requires that the number of passengers flown on each leg does not violate the capacity of the aircraft type assigned to that leg for each fare class. Note that this constraint also ensures that if $\sum_{a \in AT} x_{aj} = 0$ in (3) for any $j \in L^O$, then $\pi_{ph} = 0, \forall p \in \Pi(j)$. Constraint (7) restricts the number of passengers accepted on each path to be no more than the expected demand on that path for each fare class. Finally, constraint (8) imposes logical restrictions on the decision variables.

We assume that aircraft turn times in (4) and availabilities in (5) are suitably adjusted to permit routine scheduled maintenance requirements, given any feasible solution to model FSFAM1 (see Clarke et al. 1996, Talluri 1998, Gopalan and Talluri 1998). To be more specific, if maintenance requires a short period of time, for example, on a daily basis, it is conducted following each flight and built into the turn times with respect to constraint (4). This can be realized by linking any maintenance requirement with a particular flight leg. Furthermore, maintenance requiring four to five hours every three to five days (type A) can be assumed to be conducted overnight, as is typically done in practice when short-haul aircraft are inactive. This would involve requiring a minimum number of aircraft of each type to overnight at a maintenance station. For this purpose, as in Barnhart et al. (1998) or Clarke et al. (1996), a maintenance arc or *leapfrog arc* can be generated connecting an end of day flight arrival time at a maintenance station to a ground arc node for each aircraft type, with an associated time equal to the maintenance time. On the other hand, more extensive scheduled maintenance activities that require aircraft to be taken out of service for a long period of time can be accommodated by appropriately decreasing the number of available aircraft of each type with respect to constraint (5). However, because the fleet assignment model assigns aircraft types, instead of individual aircraft, to flight legs, it can only provide a desired number of maintenance opportunities but cannot ensure that the interval between maintenance visits can subsequently be appropriately spaced for each aircraft. Although aggregate maintenance constraints as discussed above are typically incorporated within the fleet assignment submodel, to assure a proper maintenance schedule, it is necessary to construct an actual

routing of individual aircraft, which is usually conducted subsequently at the aircraft routing step.

Although FSFAM1 is a mathematically correct model, it can be significantly improved by incorporating the z -variables defined above along with an accompanying set of constraints that link the z -variables to the (x, π) -variables. As we shall see in our computational experiments, this augmentation (as well as the further polyhedral analysis conducted in §3) substantially enhances the solvability of the model by tightening its linear programming (LP) relaxation. Toward this end, consider the following constraints:

$$z_p - \sum_{a \in AT} x_{aj} \leq 0, \quad \forall p \in \Pi^O, \forall j \in L^O(p), \quad (9)$$

$$z_p - \sum_{a \in AT} \sum_{j \in L^O(p)} x_{aj} \geq 1 - |L^O(p)|, \quad \forall p \in \Pi^O, \quad (10)$$

$$\sum_{a \in AT} x_{aj} \leq \sum_{p \in \Pi^O(j)} z_p, \quad \forall j \in L^O, \quad (11)$$

$$\pi_{ph} \leq \mu_{ph} z_p, \quad \forall p \in \Pi^O, \forall h \in H_p, \quad (12)$$

$$\pi_{ph} \leq \mu_{ph}, \quad \forall p \in \Pi \setminus \Pi^O, \forall h \in H_p, \quad (13)$$

$$(x, z): \text{ binary}, \quad (w, \pi) \geq 0. \quad (14)$$

Constraint (9) ensures that if any leg is excluded from the network, then all paths that contain this leg will also be excluded; constraint (10) ensures that if all the legs contained in a path are included in the network, then the path that contains them is also included. Constraint (11) relates the optional legs to the corresponding paths, asserting that if an optional leg is selected, then at least one path that includes this optional leg must also be activated. In context, this complements constraint (9), which implies that if an optional path is activated, then an aircraft must be assigned to every optional leg that belongs to this path. In an algorithmic process that decomposes the problem by considering fixed values of the z_p -variables as we shall adopt, such a valid inequality (11) can help reduce the number of corresponding surviving x_{aj} -variables in the resulting subproblem. Constraints (12) and (13) partition (7) into restrictions for optional and mandatory paths, respectively, tightening this relationship for the optional paths by asserting that if $z_p = 0$ for any $p \in \Pi^O$, then no demand can be accepted on this path. Finally, (14) replaces (8) to include the binary restrictions on the z -variables. We shall refer to the model defined by (1)–(6) and (9)–(14) as FSFAM2, and we proceed now to further analyze and improve this augmented model.

3. Valid Inequalities via a Polyhedral Analysis

In this section, we tighten certain existing constraints within model FSFAM2, and further propose the

generation of additional valid inequalities via suitable separation problems and partial convex hull characterizations, also exploring their facet-defining properties. Our computational results in §5 strongly justify the utility of these valid inequalities by demonstrating that they help to significantly improve the performance of the proposed algorithm.

3.1. Tightening Inequalities Within Model FSFAM2

This section addresses the lifting of constraints (6) and (12) within model FSFAM2 via Propositions 1 and 2 below, respectively.

PROPOSITION 1. *The following is a valid inequality that can be used to replace constraint (6):*

$$\sum_{p \in \Pi(j)} \pi_{ph} \leq \sum_{a \in AT} \widetilde{\text{Cap}}_{ahj} x_{aj}, \quad \forall j \in L, h \in H, \quad (15)$$

where

$$\widetilde{\text{Cap}}_{ahj} \equiv \min \left\{ \text{Cap}_{ah}, \sum_{p \in \Pi(j)} \mu_{ph} \right\}, \quad \forall a \in AT, h \in H, j \in L.$$

PROOF. Consider any feasible solution $(\bar{x}, \bar{z}, \bar{w}, \bar{\pi})$ to model FSFAM2 and examine any $j \in L, h \in H$. If $\sum_{a \in AT} \bar{x}_{aj} = 0$ (for the case of $j \in L^0$), then (15) is valid by (6). Else, we have $\bar{x}_{a^*j} = 1$ for some $a^* \in AT$ and $\bar{x}_{aj} = 0, \forall a \in AT, a \neq a^*$. In this case, (6) implies that $\sum_{p \in \Pi(j)} \bar{\pi}_{ph} \leq \text{Cap}_{a^*h}$ and (12) and (13) imply that $\sum_{p \in \Pi(j)} \bar{\pi}_{ph} \leq \sum_{p \in \Pi(j)} \mu_{ph}$, which yields $\sum_{p \in \Pi(j)} \bar{\pi}_{ph} \leq \text{Cap}_{a^*hj}$, or that (15) is again valid. Moreover, (15) implies (6), even in the continuous sense, and hence can be used to replace it. \square

Henceforth, we shall assume that (6) has been replaced by the tighter inequality (15), which can particularly be useful when capacity exceeds demand in the sense that $\text{Cap}_{ah} > \sum_{p \in \Pi(j)} \mu_{ph}$ for some $a \in AT, h \in H, j \in L$. On the other hand, when demand exceeds capacity in the sense that $\mu_{ph} > \max_{a \in AT} \text{Cap}_{ah}$ for some $p \in \Pi^0, h \in H_p$, then the following result offers a tightening of (12).

PROPOSITION 2. *The following valid inequalities can be used to replace constraint (12):*

$$\pi_{ph} \leq \tilde{\mu}_{ph} z_p, \quad \forall p \in \Pi^0, h \in H_p, \quad (16)$$

where

$$\tilde{\mu}_{ph} \equiv \min \left\{ \mu_{ph}, \max_{a \in AT} \text{Cap}_{ah} \right\}, \quad \forall p \in \Pi^0, h \in H_p.$$

PROOF. For any feasible solution $(\bar{x}, \bar{z}, \bar{w}, \bar{\pi})$ to model FSFAM2 and for any $p \in \Pi^0, h \in H_p$, if $\bar{z}_p = 0$, then (16) is valid by (12), and if $\bar{z}_p = 1$, then (12) implies that $\bar{\pi}_{ph} \leq \mu_{ph}$, and (6) along with (2) and (3) implies that $\bar{\pi}_{ph} \leq \max_{a \in AT} \text{Cap}_{ah}$, so that (16) is again valid. \square

3.2. Additional Valid Inequalities and Separation Routines

We now propose certain additional classes of valid inequalities where suitable members of such classes can be generated via separation routines as discussed below to further tighten the model representation.

PROPOSITION 3. *The following are valid inequalities for model FSFAM2:*

$$\pi_{ph} \leq \sum_{a \in AT} \min\{\mu_{ph}, \text{Cap}_{ah}\} x_{aj}, \quad \forall p \in \Pi(j), j \in L, h \in H_p. \quad (17)$$

PROOF. Let $(\bar{x}, \bar{z}, \bar{w}, \bar{\pi})$ be any feasible solution to model FSFAM2 and examine any $p \in \Pi(j), j \in L, h \in H_p$. If $\sum_{a \in AT} \bar{x}_{aj} = 0$, then (17) is valid from (6). Otherwise, from (2) and (3) we have that $\bar{x}_{a^*j} = 1$ for some $a^* \in AT$ and $\bar{x}_{aj} = 0, \forall a \in AT, a \neq a^*$. In this case, (6) implies that $\bar{\pi}_{ph} \leq \text{Cap}_{a^*h}$ and (12) and (13) imply that $\bar{\pi}_{ph} \leq \mu_{ph}$, which yields $\bar{\pi}_{ph} \leq \min\{\mu_{ph}, \text{Cap}_{a^*h}\}$ or that (17) holds true. \square

The next class of valid inequalities lifts (10) for any $p \in \Pi^0$ based on an alternative optional path $q \in \Pi^0$ that shares at least two optional legs with it.

PROPOSITION 4. *The following are valid inequalities for model FSFAM2:*

$$z_p - \sum_{a \in AT} \sum_{j \in L^0(p) \setminus L^0(q)} x_{aj} \geq z_q - |L^0(p) \setminus L^0(q)|, \quad \forall p \in \Pi^0, q \in \Pi^0, |L^0(p) \cap L^0(q)| \geq 2. \quad (18)$$

Moreover, (18) is implied by (9), (10), and $0 \leq z_q \leq 1$ when $|L^0(p) \cap L^0(q)| \leq 1$.

PROOF. When $z_q = 0$ or $\sum_{a \in AT} x_{aj} = 0$ for any $j \in L^0(p) \setminus L^0(q)$, then (18) is implied by (3) and $z_p \geq 0$. When $z_q = 1$ and $\sum_{a \in AT} x_{aj} = 1, \forall j \in L^0(p) \setminus L^0(q)$, then (18) asserts that $z_p = 1$, which is again valid. Hence, (18) is valid for all $p \in \Pi^0, q \in \Pi^0$. However, when $|L^0(p) \cap L^0(q)| \leq 1$, we have from (10) and (9) that

$$\begin{aligned} z_p - \sum_{a \in AT} \sum_{j \in L^0(p) \setminus L^0(q)} x_{aj} &\geq \sum_{a \in AT} \sum_{j \in L^0(p) \cap L^0(q)} x_{aj} + 1 - |L^0(p)| \\ &\geq z_q |L^0(p) \cap L^0(q)| + 1 - |L^0(p)| \\ &= [z_q - |L^0(p) \setminus L^0(q)|] \\ &\quad + [1 - |L^0(p) \cap L^0(q)|](1 - z_q) \\ &\geq z_q - |L^0(p) \setminus L^0(q)|, \end{aligned}$$

where the first inequality simply rewrites (10), the second inequality uses (9) (written for $q \in \Pi^0$), and the final inequality follows from the nonnegativity of the second term in the preceding equation. Hence, (18) is implied by the continuous relaxation to FSFAM2 in this case. \square

REMARK 1. It is not likely useful to incorporate all the inequalities (17) or (18) a priori within model FSFAM2. Rather, we can solve the LP relaxation of model FSFAM2 and then include members of (17) and (18) that are violated at the resulting solution, if any, within the model at the root node before proceeding further with the algorithmic process.

Next, prompted by (9) and (11), we describe another class of valid inequalities. Let

$$L^{O+} \equiv \{j \in L^O : |L^O(p)| \geq 2, \forall p \in \Pi^O(j)\},$$

and for any $j^* \in L^{O+}$, define

$$S_p^{j^*} \equiv L^O(p) \setminus \{j^*\}, \forall p \in \Pi^O(j^*)$$

and compute

$$\nu^{j^*} = \min_{p \in \Pi^O(j^*)} |S_p^{j^*}|. \quad (19)$$

Note that because $j^* \in L^{O+}$, we have $\nu^{j^*} \geq 1$.

PROPOSITION 5. For any $j^* \in L^{O+}$, let $S^{j^*} \subseteq S_U^{j^*} \equiv \bigcup_{p \in \Pi^O(j^*)} S_p^{j^*}$ be such that $|S^{j^*} \cap S_p^{j^*}| \geq \nu^{j^*}, \forall p \in \Pi^O(j^*)$. Then the following is a valid inequality:

$$\sum_{j \in S^{j^*}} \sum_{a \in AT} x_{aj} \geq \nu^{j^*} \sum_{a \in AT} x_{aj^*}. \quad (20)$$

PROOF. Note that if $\sum_{a \in AT} x_{aj^*} = 0$ then (20) is trivially valid. Else, from (3), we have that $\sum_{a \in AT} x_{aj^*} = 1$, and so from (11), there exists a $p^* \in \Pi^O(j^*)$ for which $z_{p^*} = 1$. This in turn implies from (3) and (9) that $\sum_{a \in AT} x_{aj} = 1, \forall j \in S_{p^*}^{j^*}$. Because $|S^{j^*} \cap S_{p^*}^{j^*}| \geq \nu^{j^*}$ by the given hypothesis, we therefore have (20) holding true. \square

To generate a strong version of (20) for any given $j^* \in L^{O+}$, where ν^{j^*} is then computed via (19), we would like to utilize a set S^{j^*} as per Proposition 5 for which $|S^{j^*}|$ is as small as possible. This can be accomplished by preferentially selecting legs to include within S^{j^*} that repeatedly appear within the sets $S_p^{j^*}$ for $p \in \Pi^O(j^*)$. With this motivation, a routine was designed to a priori generate inequalities of the type (20) (see the Online Supplement available at <http://joc.pubs.informs.org/ecompanion.html>); however, our computations revealed that a better alternative was to solve the following separation problem to generate a valid inequality (20) that deletes a computed optimal solution $(\bar{x}, \bar{z}, \bar{w}, \bar{\pi})$ to the LP relaxation of model FSFAM2, if possible. Toward this end, for a selected $j^* \in L^{O+}$ and its corresponding value ν^{j^*} given by (19), let

$$\theta \equiv \nu^{j^*} \sum_{a \in AT} \bar{x}_{aj^*}. \quad (21)$$

We would now like to select $S^{j^*} \subseteq S_U^{j^*}$ as per Proposition 5 that minimizes the left-hand side of (20) for the current LP solution. Hence, defining binary variables

$$y_j = \begin{cases} 1, & \text{if } j \in S^{j^*} \\ 0, & \text{otherwise} \end{cases} \quad \forall j \in S_U^{j^*},$$

we can formulate the following separation problem (SEP1) to generate (20), where the objective function determines the left-hand side of (20) at the given LP solution, and the constraints represent the restrictions on S^{j^*} as per Proposition 5.

$$(SEP1) \quad \text{Minimize} \quad \sum_{j \in S_U^{j^*}} \left[\sum_{a \in AT} \bar{x}_{aj} \right] y_j \quad (22)$$

$$\text{subject to} \quad \sum_{j \in S_p^{j^*}} y_j \geq \nu^{j^*}, \quad \forall p \in \Pi^O(j^*), \quad (23)$$

$$y: \text{ binary}. \quad (24)$$

If the optimal objective value to problem SEP1 (or any heuristic feasible solution value) is less than θ as given by (21), then the corresponding cut (20) generated via the corresponding y -solution will delete the current LP solution. Several rounds of such cuts (based on different $j^* \in L^{O+}$) can be generated and appended to the model.

EXAMPLE 1. Suppose that for a given $j^* = 1$ and $\Pi^O(j^*) = \{1, 2, 3, 4\}$, with $S_1^{j^*} = \{5, 6\}$, $S_2^{j^*} = \{5, 7, 8, 9\}$, $S_3^{j^*} = \{7, 10, 12\}$, and $S_4^{j^*} = \{6, 7, 15\}$, we have $\sum_{a \in AT} \bar{x}_{aj^*} = 1$, with $\bar{z}_p = 1/4, \forall p = 1, \dots, 4$, and that $\sum_{a \in AT} \bar{x}_{aj} = 1/4, \forall j \in S_U^{j^*}$. Hence, this continuous solution satisfies the relevant subset of constraints (3) and (9)–(11) in model FSFAM2. Note that we have $\theta = \nu^{j^*} = 2$ and $S_U^{j^*} = \{5, 6, 7, 8, 9, 10, 12, 15\}$ in this example. The separation problem (22)–(24) is then given as follows:

$$(SEP1) \quad \text{Minimize} \left\{ \sum_{j \in S_U^{j^*}} (1/4)y_j; y_5 + y_6 \geq 2; \right. \\ \left. y_5 + y_7 + y_8 + y_9 \geq 2; y_7 + y_{10} + y_{12} \geq 2; \right. \\ \left. y_6 + y_7 + y_{15} \geq 2; y: \text{ binary} \right\}.$$

An optimal solution to this problem is given by $y_5^* = y_6^* = y_7^* = y_{10}^* = 1$, and $y_j^* = 0, \forall j \in S_U^{j^*} \setminus \{5, 6, 7, 10\}$, with the optimal objective value being 1, which is less than $\theta = 2$. Hence, y^* generates the same cut (see the Online Supplement) as for Example 1, which happens to delete the given LP solution.

The following section explores conditions under which (20) is facet defining with respect to a particular substructure of model FSFAM2.

3.3. Partial Convex Hull Representations and Related Facets and Separation Routines

Given any $j^* \in L^{O+}$, define ν^{j^*} as in (19). Consider the set of constraints defined by (3), (9), and (11). Noting the appearance of the x -variables in these constraints, let us define the aggregate variables

$$\xi_j \equiv \sum_{a \in AT} x_{aj}, \quad \forall j \in L^O. \quad (25)$$

Now, for the given j^* , consider the substructure defined by (3), (9), and (11) under the change of variables (25) as given by

$$z_p - \xi_j \leq 0, \quad \forall p \in \Pi^O(j^*), \forall j \in L^O(p) \quad (26)$$

$$\xi_{j^*} \leq \sum_{p \in \Pi^O(j^*)} z_p \quad (27)$$

$$0 \leq \xi_j \leq 1, \quad \forall j \in L^O(p), p \in \Pi^O(j^*), \quad (28)$$

$$z_p \text{ binary}, \quad \forall p \in \Pi^O(j^*).$$

Note that by virtue of (3), (14), and (25), although the ξ_j -variables are binary valued, we have declared them to be continuous on $[0, 1]$ in (26)–(28) because it is easily verified that these variables are automatically binary valued at extreme point solutions to (26)–(28) for any fixed binary values for $z_p, \forall p \in \Pi^O(j^*)$. Also, observe that if $\xi_{j^*} = 0$, then $z_p = 0, \forall p \in \Pi^O(j^*)$ by (26) because $j^* \in L^O(p), \forall p \in \Pi^O(j^*)$ by the definition of $\Pi^O(j^*)$. Moreover, if $\xi_{j^*} = 1$, then the system (26)–(28) asserts that $\{\xi_j \geq 1, \forall j \in S_p^{j^*}\}$ holds true for at least one $p \in \Pi^O(j^*)$. Hence, we can equivalently restate (26)–(28) as the following disjunction in terms of just the ξ_j -variables:

$$\bigvee_{p \in \Pi^O(j^*)} \{ \xi_j \geq \xi_{j^*}, \forall j \in S_p^{j^*}; 0 \leq \xi_j \leq 1, \forall j \in S_p^{j^*} \cup \{j^*\} \}, \quad (29)$$

where we have explicitly written $S_p^{j^*} \cup \{j^*\}$ in lieu of $L^O(p)$ for the sake of clarity.

PROPOSITION 6. The convex hull of (29) is given by

$$C(j^*) = \left\{ (\xi_j, \quad j \in S_U^{j^*} \cup \{j^*\}): \right.$$

$$\xi_j^p \geq \xi_{j^*}^p, \quad \forall j \in S_p^{j^*}, \forall p \in \Pi^O(j^*), \quad (30)$$

$$0 \leq \xi_j^p \leq \lambda_p, \quad \forall j \in S_p^{j^*} \cup \{j^*\}, \forall p \in \Pi^O(j^*), \quad (31)$$

$$\sum_{p \in \Pi^O(j^*)} \lambda_p = 1, \quad (32)$$

$$\xi_j = \sum_{p \in \Pi^O(j^*)} \xi_j^p, \quad \forall j \in S_U^{j^*} \cup \{j^*\}. \quad (33)$$

PROOF. Follows directly from Balas (1998) (or see Sherali and Shetty 1980). \square

PROPOSITION 7. The facet-defining inequalities (or simply, facets) of $C(j^*)$ are of the type

$$\sum_{j \in S_U^{j^*}} \gamma_j \xi_j \geq \gamma_{j^*} \xi_{j^*} + \gamma_o, \quad (34)$$

where the vector (γ, γ_o) along with the vector (α, β) correspond to extreme directions of the following pointed polyhedral cone:

$$\gamma_j \geq \alpha_j^p - \beta_j^p, \quad \forall j \in S_p^{j^*}, p \in \Pi^O(j^*), \quad (35)$$

$$\gamma_{j^*} \leq \sum_{j \in S_p^{j^*}} \alpha_j^p + \beta_{j^*}^p, \quad \forall p \in \Pi^O(j^*), \quad (36)$$

$$\sum_{j \in S_p^{j^*} \cup \{j^*\}} \beta_j^p + \gamma_o = 0, \quad \forall p \in \Pi^O(j^*), \quad (37)$$

$$(\alpha, \beta) \geq 0. \quad (38)$$

PROOF. See the Online Supplement.

REMARK 2. Propositions 6 and 7 can be used in one of two ways (or in a combination of these two strategies as discussed below). First, we could directly include the partial convex hull representation $C(j^*)$ given by Proposition 6 (with the ξ_j -variables replaced by the x -variables using (25)) for some particular indices $j^* \in L^{O+}$, e.g., those for which the disjunction (29) is violated at the optimum obtained for the LP relaxation. Alternatively, given \bar{x} as this LP relaxation solution, we can compute $\bar{\xi}$ from (25), and if this $\bar{\xi}$ violates any disjunction given by (29), we can solve the following separation problem to possibly delete \bar{x} , based on Proposition 7:

$$(SEP2) \text{ Minimize } \sum_{j \in S_U^{j^*}} \bar{\xi}_j \gamma_j - \bar{\xi}_{j^*} \gamma_{j^*} - \gamma_o \quad (39)$$

$$\text{subject to } (35)–(38) \quad (40)$$

$$\sum_{j \in S_U^{j^*}} \bar{\xi}_j \gamma_j - \bar{\xi}_{j^*} \gamma_{j^*} - \gamma_o \geq -1, \quad (41)$$

where (41) is a normalization constraint added by way of bounding SEP2. Whenever SEP2 yields an optimum having a negative objective value (which would then equal -1 by virtue of (41)), we will have actually generated a facet of $C(j^*)$ (by Proposition 7), which deletes the current LP solution.

Finally, we address the issue raised in Remark 1 and identify certain sufficient conditions under which the valid inequality (20) given by Proposition 5 would be facet-defining for $C(j^*)$ as per Proposition 7.

PROPOSITION 8. Consider the inequality (20) given by Proposition 5, which under (25), is restated as follows:

$$\sum_{j \in S_U^{j^*}} \xi_j \geq \nu^{j^*} \xi_{j^*}. \quad (42)$$

Suppose that

$$|S^j \cap S_p^j| = \nu^j, \quad \forall p \in \Pi^O(j^*) \quad \text{and that} \quad (43)$$

$$|S^j| = |\Pi^O(j^*)|.$$

Moreover, suppose that the following equations are linearly independent:

$$\sum_{j \in S_p^j \cap S^j} \gamma_j = \nu^j, \quad \forall p \in \Pi^O(j^*). \quad (44)$$

Then (42) defines a facet of $C(j^*)$.

PROOF. See the Online Supplement.

EXAMPLE 2. Consider the data of Example 1 and the inequality (20) generated via Routine R1 (see the Online Supplement), which in the form of (42), is given by

$$\xi_5 + \xi_6 + \xi_7 + \xi_{10} \geq 2\xi_1, \quad (45)$$

where $S^j \equiv \{5, 6, 7, 10\}$ and $\nu^j = 2$. Furthermore, (43) holds true, and (44) is given by $\gamma_5 + \gamma_6 = 2$, $\gamma_5 + \gamma_7 = 2$, $\gamma_7 + \gamma_{10} = 2$, and $\gamma_6 + \gamma_7 = 2$, which uniquely yields $\gamma_5 = \gamma_6 = \gamma_7 = \gamma_{10} = 1$. Hence, by Proposition 8, (45) is facet-defining for $C(1)$. \square

4. Solution Algorithms

We now propose two algorithmic approaches for optimizing model FSFAM2. The first (denoted Algorithm A1 below) begins by tightening the representation of model FSFAM2 using the valid inequalities proposed in §3 and then applies Benders' decomposition to a suitable relaxation of this lifted model in order to prescribe the set of optional legs to include within the schedule. Finally, the resulting fleet assignment problem is solved after fixing the optional leg selections as determined above. Note that the advantage of adopting this decomposed approach, aside from making the model more computationally tractable, is that having ascertained the complete schedule at the final step, we can reuse the implication this has on demands as in Lohatepanont and Barnhart (2004) by using an appropriate schedule evaluation package to readjust the μ -parameters, and we can then reiterate this process as needed.

Algorithm A1

Step A1.1. *Tighten Model Representation.* We begin by tightening model FSFAM2 by replacing (6) and (12) with (15) and (16) using Propositions 1 and 2, respectively, and adding cuts of the type (20) via Routine R1. Next, we solve the LP relaxation of resulting model and use rounds of cuts given by Propositions 3–7, using the separation problems SEP1 and SEP2 for the cases of Propositions 5 and 6–7, respectively, as well as possibly including the higher-dimensional partial convex hull representations of Proposition 6.

Using $\xi \equiv (\xi_j, j \in L^O)$ as defined in (25), let us denote the cuts of the type (20) and (34) generated via SEP1 and SEP2, respectively, jointly as $A\xi \leq b$. Also, denote the generated valid inequalities of the type (17) via Proposition 3 as $Dx + E\pi \leq 0$.

Then, we can rewrite the enhanced model FSFAM2 as follows:

(FSFAM⁺)

$$\text{Maximize} \quad \sum_{p \in \Pi} \sum_{h \in H_p} f_{ph} \pi_{ph} - \sum_{a \in AT} \sum_{j \in L} c_{aj} x_{aj}$$

subject to

$$\sum_{a \in AT} x_{aj} = 1, \quad \forall j \in L^M, \quad (46)$$

$$\sum_{a \in AT} x_{aj} \leq 1, \quad \forall j \in L^O,$$

$$\sum_{j \in L} b f_{jn} x_{aj} + \sum_{g \in G_a} b g_{gn} w_g = 0, \quad \forall n \in N_a, \forall a \in AT,$$

$$\sum_{j \in CS_a} x_{aj} + \sum_{g \in CS_a} w_g \leq N A_a, \quad \forall a \in AT, \quad (47)$$

$$z_p - \sum_{a \in AT} x_{aj} \leq 0, \quad \forall p \in \Pi^O, \forall j \in L^O(p),$$

$$z_p - \sum_{a \in AT} \sum_{j \in L^O(p)} x_{aj} \geq 1 - |L^O(p)|, \quad \forall p \in \Pi^O,$$

$$\sum_{a \in AT} x_{aj} \leq \sum_{p \in \Pi^O(j)} z_p, \quad \forall j \in L^O,$$

$$\sum_{p \in \Pi(j)} \pi_{ph} \leq \sum_{a \in AT} \widetilde{Cap}_{ahj} x_{aj}, \quad \forall j \in L, \forall h \in H,$$

$$\pi_{ph} \leq \tilde{\mu}_{ph} z_p, \quad \forall p \in \Pi^O, \forall h \in H_p,$$

$$\pi_{ph} \leq \mu_{ph}, \quad \forall p \in \Pi \setminus \Pi^O, \forall h \in H_p,$$

$$A\xi \leq b,$$

$$Dx + E\pi \leq 0,$$

$$\xi_j = \sum_{a \in AT} x_{aj}, \quad \forall j \in L^O, \quad (48)$$

$$(x, z): \text{ binary}, \quad (w, \pi) \geq 0.$$

Step A1.2. *Apply Benders' Decomposition to a Relaxation of Model FSFAM⁺.* Let model $\overline{\text{FSFAM}}^+$ denote model FSFAM⁺ in which the binary restrictions on the x -variables are relaxed but the ξ -variables are then explicitly required to be binary valued (actually, as shown in Proposition 9 in the Online Supplement, it is sufficient to simply restrict $\xi_j \leq 1, \forall j \in L^O$, and ξ will automatically turn out to be binary valued for any feasible solution). Hence, (46) can then be eliminated in light of (48).

We next apply Benders' decomposition to model $\overline{\text{FSFAM}}^+$, where the variables $(z, \xi) \equiv (z_p, \forall p \in \Pi^O; \text{ and } \xi_j, \forall j \in L^O)$ are used in the master program as

binary restricted variables (as mentioned above, the ξ -variables are equivalently relaxed to satisfy $0 \leq \xi_j \leq 1$, $\forall j \in L^O$), and the variables (x, w, π) are used in the subproblem as continuous variables. (See the Online Supplement for explicit formulations of the master program and the primal and dual subproblems.)

Step A1.3. Optimize a Restriction of Model FFSFAM⁺. Having obtained an optimal (partial) solution (z^*, ξ^*) to model $\overline{\text{FSFAM}}^+$, we next fix $z_p = z_p^*$, $\forall p \in \Pi^O$ in model FFSFAM2, label as mandatory all legs j in Π^O for which $\xi_j^* = 1$, and delete all the remaining legs in Π^O from the model (i.e., we delete the legs j in Π^O for which $\xi_j^* = 0$). Let $\hat{\Pi} \subseteq \Pi$ denote the set of paths resulting from fixing the z_p -variables, $\forall p \in \Pi^O$, and let $\hat{L} \subseteq L$ denote the set of legs resulting from fixing the ξ_j -variables, $\forall j \in L^O$. This yields the following version of the lifted model FFSFAM2 (using Proposition 1, which remains valid):

$$\text{Maximize} \quad \sum_{p \in \hat{\Pi}} \sum_{h \in H_p} f_{ph} \pi_{ph} - \sum_{a \in AT} \sum_{j \in \hat{L}} c_{aj} x_{aj} \quad (49)$$

subject to

$$\begin{aligned} \sum_{a \in AT} x_{aj} &= 1, \quad \forall j \in \hat{L}, \\ \sum_{j \in \hat{L}} b f_{jn} x_{aj} + \sum_{g \in G_a} b g_{gn} w_g &= 0, \quad \forall n \in N_a, \forall a \in AT, \\ \sum_{j \in CS_a \cap \hat{L}} x_{aj} + \sum_{g \in CS_a} w_g &\leq NA_a, \quad \forall a \in AT, \\ \sum_{p \in \hat{\Pi}(j)} \pi_{ph} &\leq \sum_{a \in AT} \widetilde{Cap}_{ahj} x_{aj}, \quad \forall j \in \hat{L}, \forall h \in H, \\ \pi_{ph} &\leq \mu_{ph}, \quad \forall p \in \hat{\Pi}, \forall h \in H_p, \\ x: \text{ binary}, \quad (w, \pi) &\geq 0. \end{aligned}$$

Algorithm A2

We adopt an identical approach to Algorithm A1, except that at step A1.2, we solve the relaxed mixed-integer program (MIP) given by model $\overline{\text{FSFAM}}^+$ directly in lieu of applying Benders' decomposition to this problem.

5. Computational Experiments

The algorithms proposed in §4 were implemented in AMPL CPLEX 10.1 on a Precision PWS690 computer having an Intel Xeon 2.33 GHz processor with 3.25 GB of RAM and running Windows XP. Two sets of computational experiments were performed. First, while limiting the computational run-time to 12 CPU hours, we compared the results of three cases; where we (a) apply neither valid inequalities nor Benders' decomposition, (b) utilize only valid inequalities, and (c) incorporate both of these features. In a second set

of runs, for the model enhanced by the proposed valid inequalities that were found to be beneficial in the foregoing experiment, we applied Benders' decomposition and studied its computational performance for different optimality gap tolerance levels.

For test purposes, we used seven data sets based on real data provided by United Airlines. These are designated as follows, with respective numbers of flights and itineraries (paths) specified within parentheses: D1 (314 flights and 4,780 paths), D2 (428 flights, 2,282 paths), D3 (572 flights, 3,646 paths), D4 (690 flights, 4,888 paths), D5 (1,016 flights and 10,726 paths), D6 (1,238 flights, 14,103 paths), and D7 (1,476 flights, 17,121 paths). All the legs (flights) used in these data sets were initially designated to be optional legs to render the problem more challenging to solve so as to adequately test the proposed solution strategies.

REMARK 3. Typically, there are several feasibility issues that arise in data sets, which are addressed as follows. First, the (selected) outgoing and incoming arcs need to be coordinated for each station to conserve flows. In context, because of the nature of hub-and-spoke network systems, the practical data sets used in the experiments show that flights occur in pairs; i.e., if there is a flight from station A to station B, then a subsequent corresponding return flight from station B to station A coexists in the data set. This facilitates feasible solutions. However, in general, to accommodate a wider set of feasible solutions, we can create deadhead arcs for each aircraft type network from the last flight node at each station to an appropriate time-advanced node for every other station where a deadhead flight can reach this station before the end of the day. Similar to the ground arcs, such deadhead arcs would have nonnegatively restricted flows but would inherit a suitable cost term in the objective function. Another feasibility issue may occur when the fleet size is insufficient to serve all the mandatory legs. In this case, we can incorporate an artificial variable in the right-hand side of constraint (5) (to represent a chartered aircraft) along with an appropriate cost or penalty in the objective function. \square

As a preliminary investigation motivated by the discussion in §2, we compared results from FFSFAM1 and FFSFAM2 using the data set D1 (with 188 mandatory legs and 126 optional legs) to assess the tightening effect of the additional constraints used in FFSFAM2. Both models produced an optimal solution, but the CPU time required for model FFSFAM1 was about 8.4 hours, whereas the CPU time for model FFSFAM2 was about 7.2 hours, thus reducing the solution time by about 15%. Next, we solved FFSFAM1, FFSFAM2, and FFSFAM⁺ using the data set D4 (with 104 mandatory legs and 586 optional legs) with a 24-hour time limit. Both FFSFAM2 and FFSFAM⁺ were

solved to optimality, with FSFAM2 consuming 22.8 hours and FSFAM⁺ taking 20.2 hours, thus demonstrating the tightening effect of the additional cuts used in FSFAM⁺. On the other hand, the solution of FSFAM1 terminated after the run-time limit with an optimality gap of 3.2%, where the objective function value of the best solution detected was 0.9% less than the optimal value found by FSFAM2 and FSFAM⁺. As a point of interest, the LP relaxation of FSFAM1 was solved in 1.7 seconds and yielded an objective value of 2,265,177.9; that of FSFAM2 required 122.7 seconds but substantially reduced the resulting upper bound by 81.3%, and FSFAM⁺ consumed 316.3 seconds, additionally tightening the LP relaxation value further by 0.84%.

5.1. Effect of Valid Inequalities and Benders' Methodology

To examine the benefits of utilizing the proposed valid inequalities and the Benders' decomposition approach, we first ran the MIP model FSFAM2 using the CPLEX 10.1 solver without the foregoing enhancements (Case I). Next, we ran the model using different sets of valid inequalities to ascertain their tightening effect and to assess which valid inequalities beneficially contribute toward reducing the computational effort (Case II using Algorithm A2). For this case, four different sets of valid inequalities were added sequentially, and are referred to as Cases II-1, II-2, II-3(a), and II-3(b). Case II-1 implements the valid inequalities (15), (16), (17), and (18) from Propositions 1, 2, 3, and 4, respectively. Case II-2 *additionally* implements the valid inequalities (20) from Proposition 5 that are generated via Routine R1 and the separation problem SEP1 (see (22)–(24)). Case II-3 investigates the cuts of Propositions 6 and 7 in addition to the valid inequalities of Case II-2, either directly including suitable partial convex hull representations (Case II-3(a)) or generating cuts via the separation problem SEP2 given by (39)–(41) (Case II-3(b)). More specifically, in generating cuts of type (17) and (18), we selected the 10 most violated cuts at each round of the LP relaxation solution. Similarly, cuts of type (20) and (34) were, respectively, generated via SEP1 and SEP2 (both solved using CPLEX) by selecting the indices j^* that yielded the 10 most violated inequalities in each case based on the LP relaxation solution. This was repeated for three rounds after re-solving the LP relaxation at the root node with the new cuts. Then CPLEX was directly used to solve the resulting augmented models. Finally, we implemented both the valid inequalities and Benders' decomposition and ran the model FSFAM⁺ to assess the effect of this strategy on the best objective function value achieved as well as on the CPU run time (Case III using Algorithm A1). All cases were run with a time limit of 12

CPU hours and the CPLEX default optimality tolerance of $\epsilon = 10^{-6}$.

Table 1 presents the results obtained. In the second column, we report the LP objective function value of FSFAM⁺ in relative terms (Rel LP) as a ratio with respect to the LP objective function value of FSFAM2, along with the corresponding actual CPU time in hours. On average, the LP relaxation of FSFAM2 took 0.08 CPU hours to solve (in comparison with 0.13 hours for FSFAM⁺). Furthermore, for ease in assessment and comparisons, Table 1 reports the best integer solutions obtained in 12 hours of computation using the different model cases and the CPU effort in relative terms as follows. For the objective value, we present the % *gap* for each case defined as $100(F^* - F)/F^*$, where F is the particular objective value attained and F^* is the best (maximum) objective value found across all cases. For CPU times, we use Case I as the baseline and present its (actual) CPU effort in hours, although for the other cases, we provide the relative CPU time (Rel CPU) as a fraction of the CPU time for Case I.

Cases I and II-1 passed the 12-hour CPU time limit for all the test instances, whereas Case III was solved within the run-time limit for instances D1–D4. The models for cases that incorporate the proposed valid inequalities performed better than that for Case I. Among these models enhanced by different valid inequalities, the best objective function value was achieved by Case II-2 for the instances D2–D5 and by Case II-3(b) for the instance D1. Hence, we used the model from Case II-2 and applied Benders' decomposition to it in Case III. Case III consumed the least CPU time on average (Rel CPU factor of 0.83) and provided the best solutions for the largest test cases D6 and D7 and a second-best quality solution on average (average % *gap* value of 1.17% versus 1.01% for Case II-2). The total number of Benders' cuts that were generated for Case III was limited to 100 based on some preliminary runs, where it was observed that the objective value improved no more than 0.01% when the number of Benders' cuts was permitted to be greater than 100. The CPU time for Case III includes the times required to generate Benders' cuts and to solve the lifted model FSFAM2 after fixing the optional legs and paths that were obtained and selected from model $\overline{\text{FSFAM}}^+$. On average, about 8%–23% legs were deleted at Step A1.2, and then the remaining legs and corresponding paths were used in Step A1.3. Applying Benders' decomposition as in Case III displays an overall advantage of obtaining good-quality solutions relatively fast in comparison with the other models and approaches, with Case II-2 being competitive in deriving improved solutions. As a point of interest, for Case II-2, the average proportions of CPU times spent in Steps A2.1, A2.2, and A2.3

Table 1 Comparative Results for a Sequential Implementation of Valid Inequalities and Benders' Decomposition

Problem	LP relax. of FSFAM ⁺		Case I	Case II-1	Case II-2	Case II-3(a)	Case II-3(b)	Case III
D1								
Rel LP	0.92	% gap	7.85	5.97	1.07	0.87	0.00	0.78
CPU(hr)	0.03	Rel CPU	12(hr)	1.00	0.85	0.92	0.83	0.61
D2								
Rel LP	0.94	% gap	8.06	3.71	0.00	1.23	1.97	1.19
CPU(hr)	0.04	Rel CPU	12(hr)	1.00	1.00	1.00	1.00	0.62
D3								
Rel LP	0.94	% gap	7.34	4.28	0.00	1.71	3.75	1.06
CPU(hr)	0.06	Rel CPU	12(hr)	1.00	1.00	1.00	1.00	0.73
D4								
Rel LP	0.95	% gap	8.61	3.45	0.00	4.01	2.96	1.92
CPU(hr)	0.09	Rel CPU	12(hr)	1.00	1.00	1.00	1.00	0.88
D5								
Rel LP	0.93	% gap	9.71	1.89	0.00	3.62	4.12	3.27
CPU(hr)	0.26	Rel CPU	12(hr)	1.00	1.00	1.00	1.00	1.00
D6								
Rel LP	0.96	% gap	9.18	5.10	1.71	5.23	3.98	0.00
CPU(hr)	0.31	Rel CPU	12(hr)	1.00	1.00	1.00	1.00	1.00
D7								
Rel LP	N/A	% gap	No integer solution	5.22	4.30	5.83	5.15	0.00
CPU(hr)		Rel CPU		1.00	1.00	1.00	1.00	1.00
Average								
Rel LP	0.94	% gap	8.46	4.23	1.01	3.21	3.13	1.17
CPU(hr)	0.13	Rel CPU	12(hr)	1.00	0.98	0.97	0.98	0.83

of Algorithm A2 were 2.3%, 59.8%, and 37.9%, respectively, and likewise, for Case III, the average proportions of CPU times spent in Steps A1.1, A1.2, and A1.3 of Algorithm A1 were 4.1%, 28.4%, and 67.5%, respectively.

Next, using the two best cases (Cases II-2 and III), we experimented with a modification of the data sets D1–D7 where 50% of the legs were set as mandatory and the remaining legs as optional in each data set. This made the problems relatively easier to solve and significantly reduced the computational effort by a factor of 1.1 for Case II-2 and by a factor of 1.5 for Case III (see Table 2) compared with the results in Table 1. On average, Case II-2 solved the problems in 10.5 hours and Case III solved the problems in 6.8 hours. Case II-2 found better solutions for the

instances D1–D4 and D6, whereas Case III achieved a better objective function value for the remaining instances, with the average % gaps for Cases II-2 and III being 0.50 and 0.85, respectively. This experiment reveals two important points: first, having a subset of legs as mandatory reduces the computational effort, because it diminishes the burden of simultaneously deciding on selecting a profitable mix of optional legs; second, Case III can be more effective in analyzing relatively larger instances.

5.2. Computational Performance Using Different Optimality Tolerances

Using the best-performing cases from the previous subsection (Cases II-2 and III), we next investigated the effect of employing different levels of the optimality tolerance $\epsilon\%$ at both the Steps 2 and 3 of Algorithms A1 and A2 on the relative percentage gap (% gap) and the CPU effort. Table 3 presents the results obtained using a 24-hour time limit for solving the data set D5, in particular, by way of illustration. Here, the baseline results refer to using $\epsilon = 10^{-6}$ ($10^{-4}\%$), for which we specify the % gap attained and the (actual) CPU time in hours for comparative purposes. The columns pertaining to using an optimality tolerance $\epsilon\%$ equal to 1%, 5%, and 10% record the % gap attained relative to the overall best-known solution and the relative CPU time as a fraction of the respective baseline CPU time.

As evident from Table 3, increasing the optimality tolerance gradually deteriorated the quality of

Table 2 Computational Results for Cases II-2 and III with 50% Optional Legs

Problem	Case II-2		Case III	
	% gap	Rel CPU	% gap	Rel CPU
D1	0.00	0.63	2.55	0.39
D2	0.00	0.72	1.32	0.40
D3	0.00	0.89	0.29	0.48
D4	0.00	0.86	1.24	0.52
D5	1.04	1.00	0.00	0.67
D6	0.00	1.00	0.35	0.73
D7	2.47	1.00	0.00	0.79
Average	0.50	0.87	0.85	0.57

Table 3 Computational Results with Different Optimality Tolerances

Problem	Optimality tolerances with baseline case							
	$\epsilon = 10^{-4}\%$		$\epsilon = 1\%$		$\epsilon = 5\%$		$\epsilon = 10\%$	
	% gap	CPU	% gap	Rel CPU	% gap	Rel CPU	% gap	Rel CPU
Case II-2	0.00	24	0.41	0.90	0.77	0.27	2.20	0.02
Case III	0.00	19	0.92	0.55	1.32	0.19	2.04	0.02

the solution produced and significantly shortened the CPU run-times. For Case II-2, the run with $\epsilon = 5\%$ reduced the CPU time relative to that with $\epsilon = 1\%$ by a factor of 3.34, and the CPU effort corresponding to $\epsilon = 10\%$ as compared with that for $\epsilon = 5\%$ was further reduced by a factor of 13.5 (from 6.48 hours to 0.48 hours). For Case III, the CPU time consumed in the run with $\epsilon = 5\%$ decreased by a factor of 2.89 relative to that with $\epsilon = 1\%$, and the CPU effort required for $\epsilon = 10\%$ was further reduced by a factor of 9.5 (0.38 hours versus 3.67 hours) compared with that for $\epsilon = 5\%$. Meanwhile, the optimality gap values for Cases II-2 and III steadily increased with an increase in the optimality tolerance, but not substantially, displaying a % gap of 2.20% and 2.04%, respectively, for $\epsilon = 10\%$. This suggests that both algorithmic options produce good-quality solutions relatively early and then slowly converge toward optimality, with Case III demonstrating an overall better efficiency with respect to the CPU effort.

5.3. Assessing the Impact of Integration

In the integrated model, the optional legs are selected to maximize profits based on a look-ahead perspective of considering the subsequent fleet assignment with respect to all activated legs. To study the impact of this look-ahead feature in the proposed integrated model, we considered the best-performing algorithm above (Case II-2), and for each test instance, denoting L^{O*} as the set of optional legs selected, we defined FAM* as the fleet assignment model that considers the mandatory legs plus the most profitable $|L^{O*}|$ optional legs. Here, the latter were selected as the first $|L^{O*}|$ legs from the list of optional legs arranged in order of potential profit given by $\max_{a \in AT} \{R_{aj} - c_{aj}\}$, where R_{aj} is the maximum possible revenue obtained by solving the bounded variable multidimensional knapsack problem:

$$\begin{aligned}
 R_{aj} = \text{Maximize} \quad & \sum_{p \in \Pi^O(j)} \sum_{h \in H_p} f_{ph} \pi_{ph} \\
 \text{subject to} \quad & \sum_{p \in \Pi^O(j)} \pi_{ph} \leq \text{Cap}_{ah}, \quad \forall h \in \bigcup_{p \in \Pi^O(j)} H_p, \\
 & 0 \leq \pi_{ph} \leq \mu_{ph}, \quad \forall p \in \Pi^O(j), h \in H_p.
 \end{aligned}$$

In addition, we constructed FAM** as a fleet assignment model based on optional legs selected via

the following enhanced 0–1 MIP model that considers itinerary-based demands and the activation of paths induced by the selected legs, where again, we restricted the number of optional legs selected to precisely $|L^{O*}|$:

$$\begin{aligned}
 \text{Maximize} \quad & \sum_{p \in \Pi^O} \sum_{h \in H_p} f_{ph} \pi_{ph} - \sum_{a \in AT} \sum_{j \in L^O} c_{aj} x_{aj} \\
 \text{subject to} \quad & (3); (6); (9); (10); (11) \\
 & \text{for } j \in L^O, h \in H; \text{ and (12);} \\
 & \sum_{j \in L^O} x_{aj} \leq \text{NA}_a, \quad \forall a \in AT; \\
 & \sum_{a \in AT} \sum_{j \in L^O} x_{aj} = |L^{O*}|; \\
 & (x, z): \text{ binary}, \quad \pi \geq 0.
 \end{aligned}$$

Table 4 presents the (relative) % gap and CPU times for Case II-2 versus FAM* and FAM**, using the larger practical test instances D5, D6, and D7.

The results in Table 4 indicate that the objective function values for both the sequentially solved models FAM* and FAM** worsened in comparison with the integrated model FFSAM+ (using Case II-2), yielding % gaps of 11.4% and 5.5%, respectively, on average. Note that the more sophisticated approach adopted by FAM** narrowed the % gap by a factor of 2.07 compared with that of FAM*. On the other hand, to find the most profitable $|L^{O*}|$ legs, about 4% of the total CPU time was used on average for the case of FAM*, whereas 26% of the total effort was used on average for the case of FAM**. Although each case reached the 24-hour time limit, the % gap between FFSAM+ and either FAM* or FAM** clearly demonstrates the advantage of solving the integrated flight scheduling and fleet assignment problems over the sequential approach. This translates to an estimated improvement in annual profits of \$28.3 million and \$13.7 million for FFSAM+ over FAM* and FAM**, respectively.

5.4. A Sequential Fixing Heuristic

In this section, we provide comparisons with a specially designed sequential fixing heuristic to assess its effect on solution quality and computational effort. This heuristic sequentially fixes the x -variables based

Table 4 Effect of Integration

Problem	FFSAM+ (Case II-2)		FAM*		FAM**	
	% gap	CPU	% gap	Rel CPU	% gap	Rel CPU
D5	0.00	24	14.03	1.00	5.28	1.00
D6	0.00	24	9.53	1.00	6.51	1.00
D7	0.00	24	10.65	1.00	4.76	1.00
Average	0.00	24	11.40	1.00	5.52	1.00

on the LP relaxations solved at Step 3 within Algorithms A1 and A2. More specifically, whenever we solve the LP relaxation at Step 3 of Algorithm A1 and A2, we fix the x -variables that have fractional values exceeding 0.9 to 1, but we keep free the x -variables that result in values of 1. We then resolve the LP relaxation and repeat this process for a maximum of 50 iterations (or until no additional fractional variables can be fixed), after which we solve the resulting problem as a mixed-integer program. For experimental purposes, we used all the data sets, D1–D7, and implemented Cases II-2 and III with and without using the aforementioned heuristic.

The results in Table 5 demonstrate that, whereas running Cases II-2 and III without applying the heuristic frequently passed the 12-hour run-time limit, the CPU effort with the heuristic sequential fixing process was reduced, on average, by 21% for Case III and by 18% for Case II-2. For the relatively larger test instances D6 and D7, Case III without the heuristic step achieved the best overall objective function value, which indicates that using the Benders' decomposition approach is effective in finding better solutions with reasonable effort as the problem size increases. In addition, using the heuristic sequential fixing step within Case III did not deteriorate the quality of the solution more than 2% on average. On the other hand, using the heuristic sequential fixing within Case II-2

provided better outcomes than otherwise by permitting this relatively more intense procedure to focus on exploring a promising subset of the feasible region within the set time limit, thereby improving the objective function value by 0.72% on average in comparison with not using the heuristic step. Overall, Case III (with or without the heuristic step) is a preferred option for solving relatively large-scale problems.

6. Conclusions

We have proposed an integrated schedule planning and fleet assignment model by simultaneously considering optional legs, itinerary-based demands, and multiple fare classes. The basic mixed-integer programming model developed was enhanced by using various valid inequalities generated through a polyhedral analysis and the construction of partial convex hull representations along with suitable separation routines, and a Benders' decomposition solution approach was designed to facilitate the solution process. Computational results were presented using real data obtained from United Airlines to demonstrate the efficacy of the modeling and algorithmic strategies as well as the benefits of integration. A comparison of the experimental results related to the original model and different levels of the enhanced model revealed that the best modeling strategy among those tested is to use Case II-2 (which utilizes a variety of five types of valid inequalities) for moderately large-sized problems and to use Case III (which further implements a Benders' decomposition approach) for relatively larger problems. In addition, Case III can be further augmented with a heuristic sequential fixing step for even larger-sized problems (in our runs, this resulted in less than a 2% deterioration in solution quality and reduced the effort by about 21%). An experiment was also conducted to assess the impact of integration by comparing the proposed integrated model with a sequential implementation in which the schedule planning is performed separately before the fleet assignment stage by selecting a limited number (the same number as designated by our model) of optional legs based on two alternative profit maximizing submodels. The results demonstrated a clear advantage of utilizing the integrated model in terms of the percent increase in profits (11.4% and 5.5% in comparison with using the latter two sequential models, which translates to an estimated increase in annual profits of \$28.3 million and \$13.7 million, respectively).

It would be of interest, and is the subject of a follow-up paper, to incorporate in this model additional features such as flexible flight times (i.e., departure time windows), schedule balance, and recapture issues. In addition, given the interdependence of

Table 5 Effect of the Sequential Fixing Heuristic for Cases II-2 and III

Problem	Case II-2		Case III	
	w/ Heuristic	w/o Heuristic	w/ Heuristic	w/o Heuristic
D1				
% gap	3.60	0.29	3.28	0.00
Rel CPU	0.75	0.85	0.54	0.61
D2				
% gap	3.11	0.00	3.86	1.19
Rel CPU	0.75	1.00	0.57	0.62
D3				
% gap	2.33	0.00	2.74	1.06
Rel CPU	0.77	1.00	0.61	0.73
D4				
% gap	1.26	0.00	2.49	1.92
Rel CPU	0.78	1.00	0.65	0.88
D5				
% gap	1.08	0.00	3.31	3.27
Rel CPU	0.83	1.00	0.70	1.00
D6				
% gap	1.13	1.71	1.15	0.00
Rel CPU	0.85	1.00	0.72	1.00
D7				
% gap	3.45	4.30	0.77	0.00
Rel CPU	0.88	1.00	0.81	1.00
Average:				
% gap	2.28	0.90	2.51	1.06
Rel CPU	0.80	0.98	0.66	0.83

the airline operations including schedule planning, fleet assignment, aircraft routing, and crew scheduling, further research is necessary for developing suitable integrated models and designing efficient and tractable solution approaches. The consideration of several such integrated airline operational models is part of our ongoing research.

Acknowledgments

This research has been supported in part by the National Science Foundation under Grant CMMI-0754236. The authors thank Debra Hoyt and Rivi Sandhu from United Airlines for providing the data and for their input to this research, as well as the area editor and two referees for their constructive and detailed comments. The third author would like to thank Fatimah Alnijris' Research Chair for Advanced Manufacturing Technology for the financial support provided for this research.

References

Abara, J. 1989. Applying integer linear programming to the fleet assignment problem. *Interfaces* 19(4) 20–28.

Balas, E. 1998. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Appl. Math.* 89(1) 3–44.

Barnhart, C., T. S. Kniker, M. Lohatepanont. 2002. Itinerary-based airline fleet assignment. *Transportation Sci.* 36(2) 199–217.

Barnhart, C., N. L. Boland, L. W. Clarke, E. L. Johnson, G. L. Nemhauser, R. G. Shenoi. 1998. Flight string models for aircraft fleet assignment. *Transportation Sci.* 32(3) 208–220.

Berge, M. E., C. A. Hopperstad. 1993. Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms. *Oper. Res.* 41(1) 153–168.

Clarke, L. W., C. A. Hane, E. L. Johnson, G. L. Nemhauser. 1996. Maintenance and crew considerations in fleet assignment. *Transportation Sci.* 30(3) 249–260.

Cordeau, J. F., G. Stojkovic, F. Soumis, J. Desrosiers. 2001. Benders' decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Sci.* 35(4) 375–388.

Desaulniers, G., J. Desrosiers, Y. Dumas, M. Solomon, F. Soumis. 1997. Daily aircraft routing and scheduling. *Management Sci.* 43(6) 841–855.

Gopalan, R., K. T. Talluri. 1998. The aircraft maintenance routing problem. *Oper. Res.* 46(2) 260–271.

Hane, C. A., C. Barnhart, E. L. Johnson, R. E. Marsten, G. L. Nemhauser, G. Sigismondi. 1995. The fleet assignment problem: Solving a large-scale integer program. *Math. Programming* 70(1–3) 211–232.

Hauari, M., H. D. Sherali, F. Z. Mansour, N. Aissaoui. 2009. Exact approaches for integrated aircraft fleet assignment and routing at Tunis-Air. *Comput. Optim. Appl.*, ePub ahead of print October 7, <http://www.springerlink.com/content/r8t1249j4366870>.

Jacobs, T. L., E. L. Johnson, B. C. Smith. 1999. O&D FAM: Incorporating passenger flows into the fleet assignment process. R. Darrow, ed. *Proc. 39th Ann. AGIFORS Sympos., New Orleans*, 128–161.

Kniker, T. 1998. Itinerary-based airline fleet assignment. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge.

Lohatepanont, M., E. L. Johnson, B. C. Smith. 1999. Schedule generation model. R. Darrow, ed. *Proc. 39th Ann. AGIFORS Sympos., New Orleans*, 185–198.

Li, D., H.-C. Huang, A. D. Morton, E.-P. Chew. 2006. Simultaneous fleet assignment and cargo routing using Benders' decomposition. *OR Spectrum* 28(3) 319–335.

Lohatepanont, M., C. Barnhart. 2004. Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Sci.* 38(1) 19–32.

Mercier, A., F. Soumis. 2007. An integrated aircraft routing, crew scheduling, and flight retiming model. *Comput. Oper. Res.* 34(8) 2251–2265.

Mercier, A., J.-F. Cordeau, F. Soumis. 2005. A computational study of Benders' decomposition for the integrated aircraft routing and crew scheduling problem. *Comput. Oper. Res.* 32(6) 1451–1476.

Papadakos, N. 2009. Integrated airline scheduling. *Comput. Oper. Res.* 36(1) 176–195.

Pilla, V. L., J. M. Rosenberger, V. C. P. Chen, B. Smith. 2008. A statistical computer experiments approach to airline fleet assignment. *IIE Trans.* 40(5) 524–537.

Rushmeier, R. A., S. A. Kontogiorgis. 1997. Advances in the optimization of airline fleet assignment. *Transportation Sci.* 31(2) 159–169.

Sandhu, R., D. Klabjan. 2007. Integrated airline fleet assignment and crew-pairing decisions. *Oper. Res.* 55(3) 439–456.

Sherali, H. D., C. M. Shetty. 1980. *Optimization with Disjunctive Constraints. Lecture Notes in Economics and Mathematical Systems*, Vol. 181. Springer-Verlag, Berlin.

Sherali, H. D., X. Zhu. 2008. Two-stage fleet assignment model considering stochastic passenger demands. *Oper. Res.* 56(2) 383–399.

Sherali, H. D., E. K. Bish, X. Zhu. 2005. Polyhedral analysis and algorithms for a demand-driven fleet assignment model for aircraft assignment. *Transportation Sci.* 39(3) 349–366.

Sherali, H. D., E. K. Bish, X. Zhu. 2006. Airline fleet assignment concepts, models, and algorithms. *Eur. J. Oper. Res.* 172(1) 1–30.

Subramanian, R., R. P. Scheff, J. D. Quillinan, D. S. Wiper, R. E. Marsten. 1994. Coldstart: Fleet assignment at Delta Air Lines. *Interfaces* 24(1) 104–120.

Talluri, K. T. 1998. The four-day aircraft maintenance routing problem. *Transportation Sci.* 32(1) 43–53.