

Integrated Airline Operations: Schedule Design, Fleet Assignment, Aircraft Routing, and Crew Scheduling

Ki-Hwan Bae

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Industrial and Systems Engineering

Hanif D. Sherali, Chair
C. Patrick Koelling
Subhash C. Sarin
Antonio A. Trani

November 18, 2010
Blacksburg, Virginia

Keywords: Airline Operations Research, Integrated Airline Operations, Schedule Design, Fleet Assignment, Aircraft Maintenance Routing, Crew Scheduling, Itinerary-based Demands, Mixed-Integer Programming, Polyhedral Analysis, Benders Decomposition, Reformulation-Linearization Technique.

Integrated Airline Operations: Schedule Design, Fleet Assignment, Aircraft Routing, and Crew Scheduling

Ki-Hwan Bae

(ABSTRACT)

Air transportation offers both passenger and freight services that are essential for economic growth and development. In a highly competitive environment, airline companies have to control their operating costs by managing their flights, aircraft, and crews effectively. This motivates the extensive use of analytical techniques to solve complex problems related to airline operations planning, which includes schedule design, fleet assignment, aircraft routing, and crew scheduling. The initial problem addressed by airlines is that of schedule design, whereby a set of flights having specific origin and destination cities as well as departure and arrival times is determined. Then, a fleet assignment problem is solved to assign an aircraft type to each flight so as to maximize anticipated profits. This enables a decomposition of subsequent problems according to the different aircraft types belonging to a common family, for each of which an aircraft routing problem and a crew scheduling or pairing problem are solved. Here, in the aircraft routing problem, a flight sequence or route is built for each individual aircraft so as to cover each flight exactly once at a minimum cost while satisfying maintenance requirements. Finally, in the crew scheduling or pairing optimization problem, a minimum cost set of crew rotations or pairings is constructed such that every flight is assigned a qualified crew and that work rules and collective agreements are satisfied. In practice, most airline companies solve these problems in a sequential manner to plan their operations, although recently, an increasing effort is being made to develop novel approaches for integrating some of the airline operations planning problems while retaining tractability. This dissertation formulates and analyzes three different models, each of which examines a composition of certain pertinent airline operational planning problems. A comprehensive fourth model is also proposed, but is relegated for future research.

In the first model, we integrate fleet assignment and schedule design 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 pertaining to multiple fare-classes. A polyhedral analysis of the proposed mixed-integer programming model is used to derive several classes of valid inequalities for tightening its representation. Solution approaches are developed by applying Benders decomposition method to the resulting lifted model, and computational experiments are conducted using real data obtained from a major U.S. airline (*United Airlines*) to demonstrate the efficacy of the proposed procedures as well as the benefits of integration. A comparison of the experimental results obtained for the basic integrated model and for its different enhanced representations reveals that the best modeling strategy among those tested is the one that utilizes a variety of five types of valid inequalities for moderately sized problems, and further implements a Benders decomposition approach for relatively larger problems. In addition, when a heuristic sequential fixing step is incorporated within the algorithm for even larger sized problems, the computational results demonstrate a less than 2% deterioration in solution quality, while reducing the effort by about 21%. We also performed an experiment to assess the impact of integration by comparing the proposed integrated model with a sequential implementation in which the schedule design is implemented separately before the fleet assignment stage based on two alternative profit maximizing submodels. The results obtained demonstrate a clear advantage of utilizing the integrated model, yielding an 11.4% and 5.5% increase in profits in comparison with using the latter two sequential models, which translates to an increase in annual profits by about \$28.3 million and \$13.7 million, respectively.

The second proposed model augments the first model with additional features such as flexible flight times (i.e., departure time-windows), schedule balance, and demand recapture considerations. Optional flight legs are incorporated to facilitate the construction of a profitable schedule by optimally selecting among such alternatives in concert with assigning the available aircraft fleet to all the scheduled legs. Moreover, network effects and realistic demand patterns are effectively represented by examining itinerary-based demands as well as multiple fare-classes. Allowing flexibility on the departure times of scheduled flight legs

within the framework of an integrated model increases connection opportunities for passengers, hence yielding robust schedules while saving fleet assignment costs. A provision is also made for airlines to capture an adequate market share by balancing flight schedules throughout the day. Furthermore, demand recapture considerations are modeled to more realistically represent revenue realizations. For this proposed mixed-integer programming model, which integrates the schedule design and fleet assignment processes while considering flexible flight times, schedule balance, and recapture issues, along with optional legs, itinerary-based demands, and multiple fare-classes, we perform a polyhedral analysis and utilize the Reformulation-Linearization Technique in concert with suitable separation routines to generate valid inequalities for tightening the model representation. Effective solution approaches are designed by applying Benders decomposition method to the resulting tightened model, and computational results are presented to demonstrate the efficacy of the proposed procedures. Using real data obtained from *United Airlines*, when flight times were permitted to shift by up to ± 10 minutes, the estimated increase in profits was about \$14.9M/year over the baseline case where only original flight legs were used. Also, the computational results indicated a 1.52% and 0.49% increase in profits, respectively, over the baseline case, while considering two levels of schedule balance restrictions, which can evidently also enhance market shares. In addition, we measured the effect of recaptured demand with respect to the parameter δ that penalizes switches in itineraries. Using values of δ that reflect 1, 50, 100, or 200 dollars per switched passenger, this yielded increases in recaptured demand that induced additional profits of 2.10%, 2.09%, 2.02%, and 1.92%, respectively, over the baseline case. Overall, the results obtained from the two schedule balance variants of the proposed integrated model that accommodate all the features of flight retiming, schedule balance, and demand recapture simultaneously, demonstrated a clear advantage by way of \$35.1 and \$31.8 million increases in annual profits, respectively, over the baseline case in which none of these additional features is considered.

In the third model, we integrate the schedule design, fleet assignment, and aircraft maintenance routing decisions, while considering optional legs, itinerary-based demands, flexible

flight retimings, recapture, and multiple fare-classes. Instead of utilizing the traditional *time-space network* (**TSN**), we formulate this model based on a *flight network* (**FN**) that provides greater flexibility in accommodating integrated operational considerations. In order to consider *through-flights* (i.e., a sequence of flight legs served by the same aircraft), we append a set of constraints that matches aircraft assignments on certain inbound legs into a station with that on appropriate outbound legs at the same station. Through-flights can generate greater revenue because passengers are willing to pay a premium for not having to change aircraft on connecting flights, thereby reducing the possibility of delays and missed baggage. In order to tighten the model representation and reduce its complexity, we apply the Reformulation-Linearization Technique (RLT) and also generate other classes of valid inequalities. In addition, since the model possesses many equivalent feasible solutions that can be obtained by simply reindexing the aircraft of the same type that depart from the same station, we introduce a set of suitable hierarchical symmetry-breaking constraints to enhance the model solvability by distinguishing among aircraft of the same type. For the resulting large-scale augmented model formulation, we design a Benders decomposition-based solution methodology and present extensive computational results to demonstrate the efficacy of the proposed approach. We explored four different algorithmic variants, among which the best performing procedure (Algorithm A1) adopted two sequential levels of Benders partitioning method. We then applied Algorithm A1 to perform several experiments to study the effects of different modeling features and algorithmic strategies. A summary of the results obtained is as follows. First, the case that accommodated both mandatory and optional through-flight leg pairs in the model based on their relative effects on demands and enhanced revenues achieved the most profitable strategy, with an estimated increase in expected annual profits of \$2.4 million over the baseline case. Second, utilizing symmetry-breaking constraints in concert with compatible objective perturbation terms greatly enhanced problem solvability and thus promoted the detection of improved solutions, resulting in a \$5.8 million increase in estimated annual profits over the baseline case. Third, in the experiment that considers recapture of spilled demand from primary itineraries to other compatible itineraries, the

different values of $\delta = 100, 50, \text{ and } 1$ (dollars per re-routed passenger) induced average respective proportions of 3.2%, 3.4%, and 3.7% in recaptured demand, resulting in additional estimated annual profits of \$3.7 million, \$3.8 million, and \$4.0 million over the baseline case. Finally, incorporating the proposed valid inequalities within the model to tighten its representation helped reduce the computational effort by 11% on average, while achieving better solutions that yielded on average an increase in estimated annual profits of \$1.4 million.

In closing, we propose a fourth more comprehensive model in which the crew scheduling problem is additionally integrated with fleet assignment and aircraft routing. This integration is important for airlines because crew costs are the second largest component of airline operating expenses (after fuel costs), and the assignment and routing of aircraft plus the assignment of crews are two closely interacting components of the planning process. Since crews are qualified to typically serve a single aircraft family that is comprised of aircraft types having a common cockpit configuration and crew rating, the aircraft fleet and routing decisions significantly impact the ensuing assignment of cockpit crews to flights. Therefore it is worthwhile to investigate new models and solution approaches for the integrated fleet, aircraft routing, and crew scheduling problem, where all of these important inter-dependent processes are handled simultaneously, and where the model can directly accommodate various work rules such as imposing a specified minimum and maximum number of flying hours for crews on any given pairing, and a minimum number of departures at a given crew base for each fleet group. However, given that the crew scheduling problem itself is highly complex because of the restrictive work rules that must be heeded while constructing viable duties and pairings, the formulated integrated model would require further manipulation and enhancements along with the design of sophisticated algorithms to render it solvable. We therefore recommend this study for future research, and we hope that the modeling, analysis, and algorithmic development and implementation work performed in this dissertation will lend methodological insights into achieving further advances along these lines.

Acknowledgments

My deepest gratitude goes to Professor Hanif D. Sherali, my advisor as well as my mentor, for his guidance, patience, and inspiration. I can unequivocally state that his expert knowledge and experience made this work possible. I am also fortunate to have Professors C. Patrick Koelling, Subhash C. Sarin, and Antonio A. Trani on my committee, and I am very thankful for their encouragement throughout this study and for the opportunity to learn from their perspectives about many things beyond academic work. I would like to note that Professor Mohamed Haouari has accompanied me with his valuable input in my educational journey, and that this work has been supported by the *National Science Foundation* (under Grant Number CMMI-0754236), for which I am also thankful. My colleague, Dr. Brian Lunday, has been a wonderful friend and an undeniable source of information and motivation, making my years at Virginia Tech more fun and memorable. Finally, without question, my family has been the biggest source of unconditional love and everlasting support throughout my life. To them, I like to dedicate this work.

Contents

- 1 Introduction** **1**
 - 1.1 Background 1
 - 1.2 Research Motivation 6
 - 1.3 Organization of this Dissertation 9

- 2 Literature Review** **10**
 - 2.1 Schedule Design 10
 - 2.2 Fleet Assignment 12
 - 2.3 Aircraft Routing 17
 - 2.4 Crew Scheduling 20
 - 2.5 Integrated Models for the Fleet Assignment Problem (FAP) 23
 - 2.6 Schedule Disruption and Recovery 26
 - 2.7 Revenue/Yield Management (RM/YM) 32

- 3 An Integrated Schedule Design and Fleet Assignment Model** **39**
 - 3.1 Introduction 39

| | | |
|----------|--|-----------|
| 3.2 | Model Description and Notation | 41 |
| 3.3 | Valid Inequalities via a Polyhedral Analysis | 46 |
| 3.3.1 | Tightening Inequalities within Model FSFAM2 | 46 |
| 3.3.2 | Additional Valid Inequalities and Separation Routines | 47 |
| 3.3.3 | Partial Convex Hull Representations and Related Facets and Separation Routines | 52 |
| 3.4 | Solution Algorithms | 57 |
| 3.5 | Computational Experiments | 64 |
| 3.5.1 | Effect of Valid Inequalities and Benders Methodology | 66 |
| 3.5.2 | Computational Performance Using Different Optimality Tolerances | 70 |
| 3.5.3 | Assessing the Impact of Integration | 71 |
| 3.5.4 | A Sequential Fixing Heuristic | 73 |
| 4 | An Integrated Schedule Design and Fleet Assignment Model | 75 |
| 4.1 | Introduction | 75 |
| 4.2 | Notation and Model Formulation | 76 |
| 4.3 | Valid Inequalities | 85 |
| 4.4 | Solution Approach | 89 |
| 4.5 | Computational Experiments | 99 |
| 4.5.1 | Effectiveness of Algorithmic Solution Approaches | 100 |
| 4.5.2 | Modeling Features | 103 |
| 5 | An Integrated Schedule Design, Fleet Assignment, and Aircraft Routing | |

| | |
|--|------------|
| Model | 114 |
| 5.1 Introduction | 114 |
| 5.2 Model Description and Notation | 116 |
| 5.2.1 Symmetry-breaking Constraints | 123 |
| 5.2.2 Aircraft Maintenance Routing Decisions | 124 |
| 5.2.3 Valid Inequalities | 127 |
| 5.3 Algorithmic Solution Approach | 132 |
| 5.4 Computational Results | 146 |
| 5.4.1 Preliminary Test of Algorithms | 147 |
| 5.4.2 Analyses of Different Modeling Features | 149 |
| 6 An Integrated Fleet Assignment, Aircraft Routing, and Crew Scheduling | |
| Model | 158 |
| 6.1 Introduction | 158 |
| 6.2 Model Description and Notation | 159 |
| 7 Conclusions | 166 |
| 7.1 Summary and Closing Remarks | 166 |
| Bibliography | 172 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | US airline passenger demand projection from 2000 to 2030 (Used with permission of U.S. DOT/Federal Aviation Administration, 2010; Source: http://www.rita.dot.gov) | 2 |
| 1.2 | Passenger and freight air transportation growth, and economic growth for 1950-2006 (Used with permission of Air Transport Association and World-watch Institute, 2009; Source: http://www.people.hofstra.edu) | 2 |
| 1.3 | Point-to-point system vs. hub-and-spoke system | 4 |
| 1.4 | Features that interact with airline scheduling | 5 |
| 1.5 | Airline operational planning process | 7 |
| 2.1 | A sample airline network with two hubs (ORD and IAD) and nine spokes (Used with permission of Ashgate, 2009; Source: <i>Airline Operations and Scheduling</i> , Bazargan, 2004). | 11 |
| 2.2 | A two-type, two-station time-space network. | 13 |
| 2.3 | Two-day routing for aircraft type B737-800 (Used with permission of Ashgate, 2009; Source: <i>Airline Operations and Scheduling</i> , Bazargan, 2004) | 18 |
| 2.4 | A pairing with duty periods, sits, and overnight rests (Used with permission of Ashgate, 2009; Source: <i>Airline Operations and Scheduling</i> , Bazargan, 2004) | 21 |

List of Tables

| | | |
|-----|---|-----|
| 3.1 | Comparative Results for a Sequential Implementation of Valid Inequalities and Benders Decomposition | 67 |
| 3.2 | Computational Results for Cases II-2 and III with 50% optional legs | 69 |
| 3.3 | Computational Results with Different Optimality Tolerances | 70 |
| 3.4 | Effect of Integration | 72 |
| 3.5 | Effect of the Sequential Fixing Heuristic for Cases II-2 and III | 74 |
| 4.1 | Comparative Results for Algorithms A0, A1, and A2 | 101 |
| 4.2 | Performance Characteristics of Algorithm A1 | 102 |
| 4.3 | Comparative Results for Different Flight Retiming Windows | 105 |
| 4.4 | Computational Results with Schedule Balance Consideration | 108 |
| 4.5 | Computational Results with Recapture Consideration | 110 |
| 4.6 | Comparative Results for the Baseline vs. the Overall Integrated Model | 112 |
| 5.1 | Comparative Results for Algorithms A1, A2, and A3 | 148 |
| 5.2 | Effect of Accommodating Through-flights | 151 |
| 5.3 | Computational Results with Different Symmetry-breaking Constraint Settings | 153 |

| | | |
|-----|--|-----|
| 5.4 | Computational Results with Recapture Consideration | 155 |
| 5.5 | Comparative Results with and without the Proposed Valid Inequalities . . . | 157 |

Chapter 1

Introduction

1.1 Background

Air transportation has been steadily growing in response to an increase in both passenger and freight traffic demands, and plays an essential role in both local and global economic development. According to U.S. Department of Transportation (DOT), air passenger demand is projected to double between 2000 and 2030, and current forecasts estimate 1.2 billion passengers by 2030 (see Figure 1.1). Also, based on statistics obtained from the Air Transport Association (ATA) (see Figure 1.2), there exists a positive correlation of growth rates among air passenger traffic, air freight traffic, and the Gross World Product (GWP) over the period 1950-2006. However, despite growth in travel demand, airlines have been faced with thin profit margins, especially after deregulation in 1978, due to the highly competitive nature of the airline business. Traditionally, air traffic has followed the pattern of the economic cycle because of the high sensitivity of demand to the current economic environment, usually reflecting an immediate decline in demand when the economy recedes, especially among high paying customers, and displaying a relatively slow recovery compared with other businesses when the economy turns around.

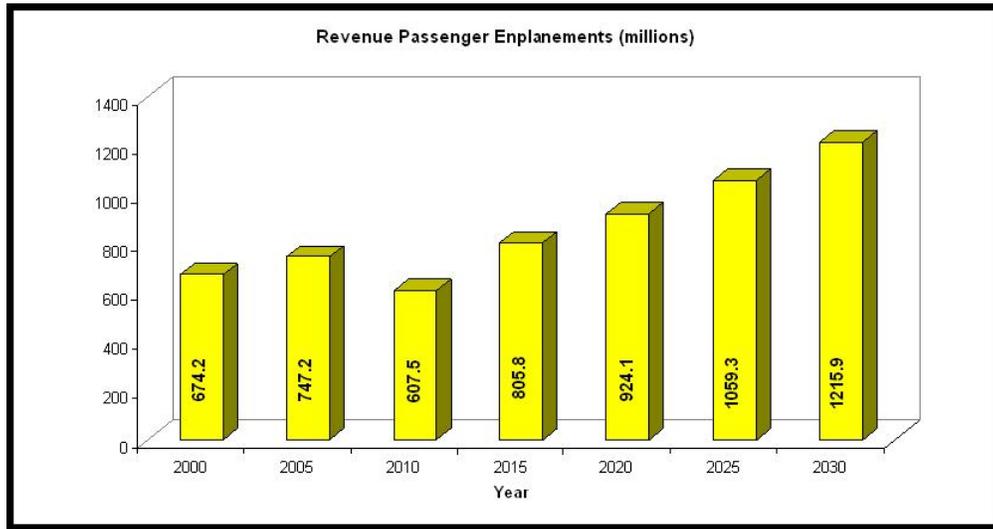


Figure 1.1: US airline passenger demand projection from 2000 to 2030 (Used with permission of U.S. DOT/Federal Aviation Administration, 2010; Source: <http://www.rita.dot.gov>)

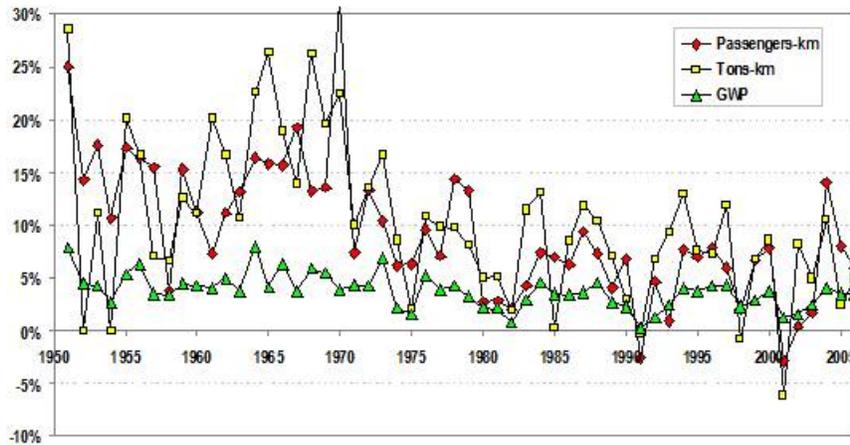


Figure 1.2: Passenger and freight air transportation growth, and economic growth for 1950-2006 (Used with permission of Air Transport Association and Worldwatch Institute, 2009; Source: <http://www.people.hofstra.edu>)

Typically, demand contributes to the characteristics of flight schedules and route networks of an airline, and governs the composition of nonstop flights or multi-leg itineraries,

origins and destinations of markets, and the departure and arrival times of related flight legs and their days of operation as well as seasonal variations. In hub-and-spoke network systems, passengers flow on the routes from spokes to hubs, between hubs, and from hubs to spokes, thereby consolidating hub operations and lowering operating unit costs by achieving economies of scale (concentration of flows), and serving more markets with a higher frequency. For example, observing Figure 1.3, a point-to-point network involves 15 independent links to connect every node to every other node, whereas it requires only five connections in a hub-and-spoke network structure to provide such a node-to-node service, implying better service efficiency and more revenue opportunities. On the other hand, a hub-and-spoke system entails the inconvenience of multiple leg trips from and to spokes, as opposed to direct services, hence increasing passenger travel times, and potentially causing delays and congestion at peak hours at hubs, and resulting in lower aircraft utilization due to waiting for connecting flights.

Airlines also manipulate or induce demand via factors such as fare-classes, available sizes of aircraft (capacity) offered, and schedule frequencies. First, to maximize profits from demand pertaining to a given flight leg, airlines use sophisticated yield management techniques to control seat inventory, saving some seats for high-paying passengers and deliberately spilling low-paying passengers on certain segments. Next, to accommodate a given demand in a particular market, airlines can choose to serve them with a larger sized aircraft operated with a relatively lower frequency, or with a smaller sized aircraft operated more often. This is also closely related to the number and schedule of flights that an airline offers in a market to win customer goodwill and loyalty. By increasing frequency, airlines can provide more attractive service to passengers by making their actual departure times closer to their desired departure times. However, this requires more resources such as crews and aircraft, and involves additional maintenance, fuel, landing fee, and gate management costs, among others.

The principal resources necessary to provide airline service to passengers are aircraft, which require long-term planning and large sums of capital to buy or lease; different grades

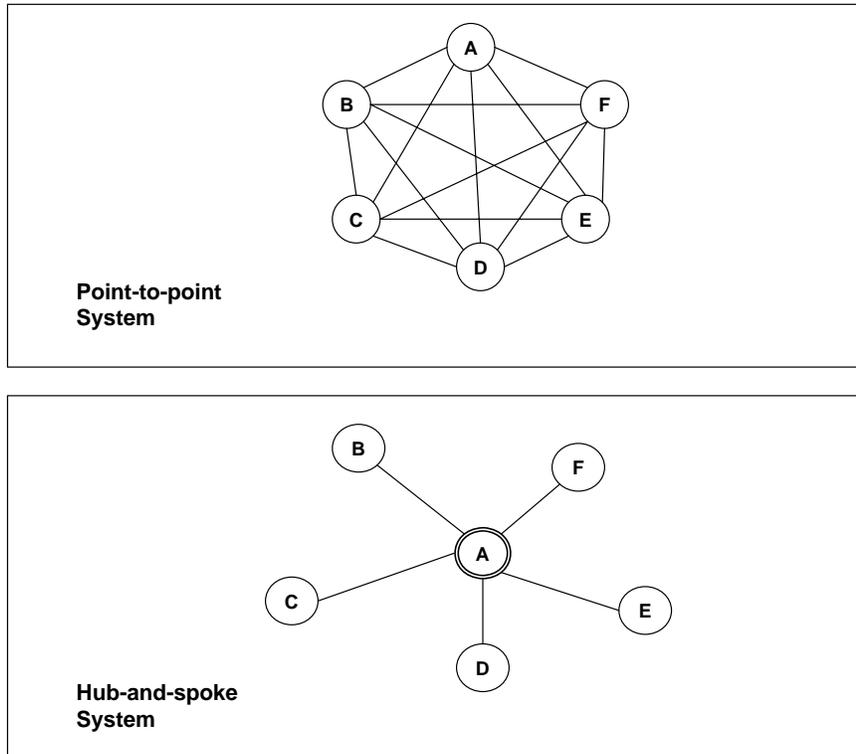


Figure 1.3: Point-to-point system vs. hub-and-spoke system

of qualified personnel, who are strictly controlled by regulations and union rules; and fuel, whose cost is volatile and directly affects the airline's marginal profitability. Thus, while considering all these elements along with the aforementioned features (see Figure 1.4), airlines need to carefully control their operating costs by judiciously managing their flights, aircraft, and crews. Recognizing that the main objective of airline scheduling is to maximize operating profits, it is important for an airline to be able to provide flight schedules that both closely match capacity with demand and, at the same time, minimize the accompanying resource allocation costs.

This motivates the extensive use of analytical techniques to solve such complex problems related to airline operations planning. In the literature, several modeling and solution ap-

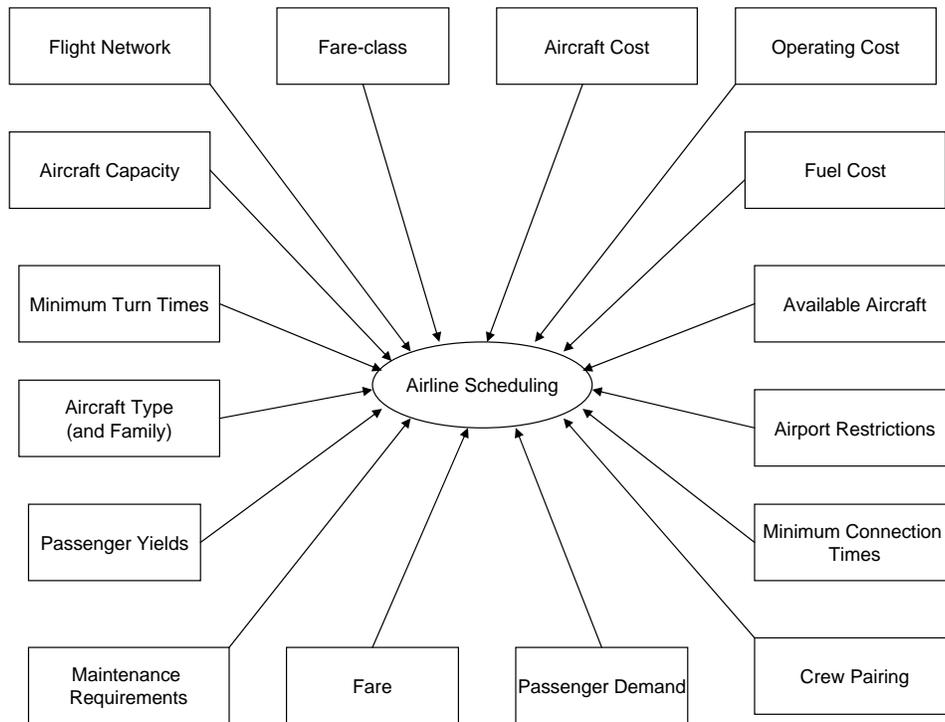


Figure 1.4: Features that interact with airline scheduling

proaches have been proposed to individually address the problems related to the different airline planning processes. In the following, an *aircraft type* will refer to a certain model of aircraft, such as the Boeing B737-500. All aircraft of the same type have the same cockpit configuration, crew rating, and capacity. An *aircraft family* is a set of aircraft types, each having the same cockpit configuration and crew rating. Thus, the same crew can fly any aircraft type of the same family.

The main airline operations consist of *schedule design*, *fleet assignment*, *aircraft routing*, and *crew scheduling*. In *schedule design*, the airline company determines a set of flights having specific origin and destination cities as well as departure and arrival times in order to generate a schedule that offers the highest potential revenue. The generated schedule

constitutes the basis of the airline's operations. Next, a *fleet assignment problem* is solved to assign aircraft types to the scheduled flights, based on equipment capacities, capabilities, availabilities, operational costs, and potential revenues. Because crews are trained to fly any aircraft type within a specified family, this enables a decomposition of subsequent problems according to the different aircraft types belonging to a common family, for each of which, an *aircraft routing problem* and a *crew scheduling* or *pairing problem* are solved. An airline's fleet decision highly impacts its revenues, and thus constitutes an essential component of its overall scheduling process. In the *aircraft routing problem*, the sequence of flight legs to be flown by each individual aircraft is determined so as to cover each flight leg exactly once at a minimum cost, while satisfying mandatory maintenance requirements. Finally, the *crew scheduling problem* is solved to construct a minimum cost set of crew rotations or pairings so that every flight is assigned a qualified crew while satisfying work rules and collective agreements. Here, a *duty period* is a single workday of a crew, and a *pairing* refers to a sequence of duty periods with overnight rests between consecutive periods. A *rotation process*, also called *rostering*, assigns individual crew members to crew pairings.

The aforementioned *schedule design* process is performed as early as one year in advance, whereas the *fleet assignment* is conducted about three months in advance, followed by the *aircraft routing* and *crew scheduling* processes that are implemented about 2-4 weeks in advance. This sequence is illustrated in Figure 1.5.

1.2 Research Motivation

Most airline companies use a sequential procedure to plan their operations. As mentioned above, the schedule design process begins 12 months prior to implementation, and involves designing flight schedules, i.e., the origin, destination, and timing of each flight. Based on this flight schedule, a flight network is constructed for solving the fleet assignment problem, whereby a fleet (aircraft type) is assigned to each flight segment, which is done about 12 weeks

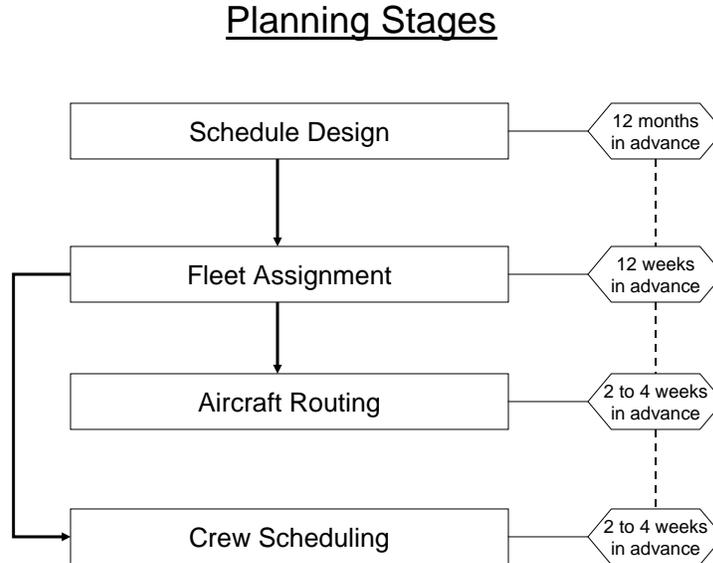


Figure 1.5: Airline operational planning process

in advance. Subsequentially, about 2-4 weeks in advance, aircraft routing and crew scheduling decisions are made. Accordingly, each problem is modeled and solved independently of the remaining ones, even though there is a clear interaction between them: the optimal solution of one problem becomes the input for the subsequent problem. This procedure considerably reduces the complexity of the process but, since the solution of one problem does not take into account the considerations that affect subsequent problems, it results in overall suboptimal solutions that might be far from optimality. Moreover, finding feasible solutions may become difficult because flexibility is diminished by previously restricted decisions.

Consequently, airline companies are actively seeking additional novel approaches to model and solve these problems *simultaneously* so as to obtain improved solutions that would lead to substantial savings over the sequential approach. Such a lower cost is an immediate necessity for any airline company participating in today's extremely competitive market.

Thus, airline operations focus on capturing the right mix of passengers for the purpose of “*putting the right plane with the right number of seats and the right crew on the right route at the right time*”.

Although integrated airline operations problems have become key to efficiently managing planes and crews for higher profits, research that focuses on the formulation and solution of such integrated problems has received relatively lesser attention due to the complexity of the considered problems, at least until very recently, and yet, there is much room for improvement. Thus, an increasing effort is being made in order to develop novel approaches for integrating some of the airline operations planning problems such as the fleet assignment and aircraft rotation problems (Barnhart et al. (1998a) and Ioachim et al. (1999)); the fleet assignment and crew pairing problems (Subramanian et al. (1994), Clarke et al. (1996), and Barnhart et al.(1998c)); and the aircraft rotation and crew pairing problems (Cordeau et al. (2001) and Mercier et al. (2005)). Recent attempts to partially integrate all the problems have been made by including constraints in a basic formulation that take into account different aspects of other correlated problems (Klabjan et al. (2002) and Sandhu and Klabjan (2007)).

Motivated by these considerations, we investigate the following integrated approaches in this dissertation for the airline operations planning process: (1) integration of schedule design and fleet assignment while considering optional legs, itinerary-based demand, and multiple fare-classes; (2) integration of schedule design and fleet assignment with optional legs, itinerary-based demands, multiple fare-classes, flight leg copies, schedule balance, and recapture considerations; and (3) integration of schedule design, fleet assignment, and aircraft routing with optional legs, itinerary-based demands, multiple fare-classes, flight leg copies, schedule balance, and recapture considerations. For each of the above, we develop a basic model, then augment it with suitable valid inequalities via a polyhedral analysis for enhancing its solvability, next design suitable solution algorithms, and finally implement and test the proposed procedures using real data obtained from *United Airlines*. In closing, we also propose a fourth model that integrates fleet assignment, aircraft routing, and

crew scheduling. Due to the complexity of this model, which would require an elaborate independent study by itself, we relegate its study for future research.

1.3 Organization of this Dissertation

This dissertation is organized as follows. In Chapter 2, we review the literature on models for each of the different airline operational planning processes, along with the existing work on integrated airline operations, as well as for airline recovery processes and related revenue/yield management. In Chapters 3-5, we propose three principal integrated approaches for airline operational planning and develop and test algorithmic solution approaches to solve each proposed model. Chapter 3 first develops an integrated schedule design and fleet assignment model, while considering optional flight legs, path/itinerary-based demands, and multiple fare-classes. Chapter 4 enhances the foregoing model by additionally considering flexible flight times, recapture issues, and schedule balance. Chapter 5 then further extends our study by incorporating the aircraft routing problem. For each of these three models, we provide a detailed polyhedral analysis, algorithmic design, and computational results in the respective chapters. As a recommendation for future research in this domain, Chapter 6 presents a more comprehensive model that integrates the fleet assignment, aircraft routing, and crew scheduling problems. Finally, Chapter 7 provides conclusions and discussions.

Chapter 2

Literature Review

2.1 Schedule Design

Schedule design, also called flight scheduling, is the basis for all the other airline planning and operations. In the schedule design stage, a flight timetable is constructed to specify what cities to fly to and from, and at what times. Various factors such as market demands, available aircraft, crew resources, regulations, airport characteristics, and schedules of other competing airlines, affect an airline's flight scheduling decisions. After deregulation in 1978, airlines have been able to choose which markets (origin and destination pairs) to serve, and the frequency of service to provide in each market. Consequently, this has led airlines to adopt hub-and-spoke operations, which yield higher revenues and efficiency compared to point-to-point operations. Typically, a major airline network consists of several hubs and each hub has a set of spokes corresponding to closely situated cities (see Figure 2.1). Usually the size of the airline network is measured by the number of airports and flight frequencies served by the airline. *Load factor* is the average rate of seats occupied by passengers in a flight and, along with frequency, is one of the measures for airlines to decide whether the operation of a particular flight segment is marketable or not. The parameters affecting load factors are origins and destinations, fare-classes, flight times, and flight frequencies. Most

airline companies use a manual procedure for schedule design, driven basically by marketing requirements. However, some recent attempts to use modeling and optimization techniques for generating improved schedules have been made. For example, Erdmann et al. (2001) have addressed a schedule generation problem that arises particularly at charter companies. This problem was modeled as a capacitated network design problem and solved using a tailored branch-and-cut algorithm.

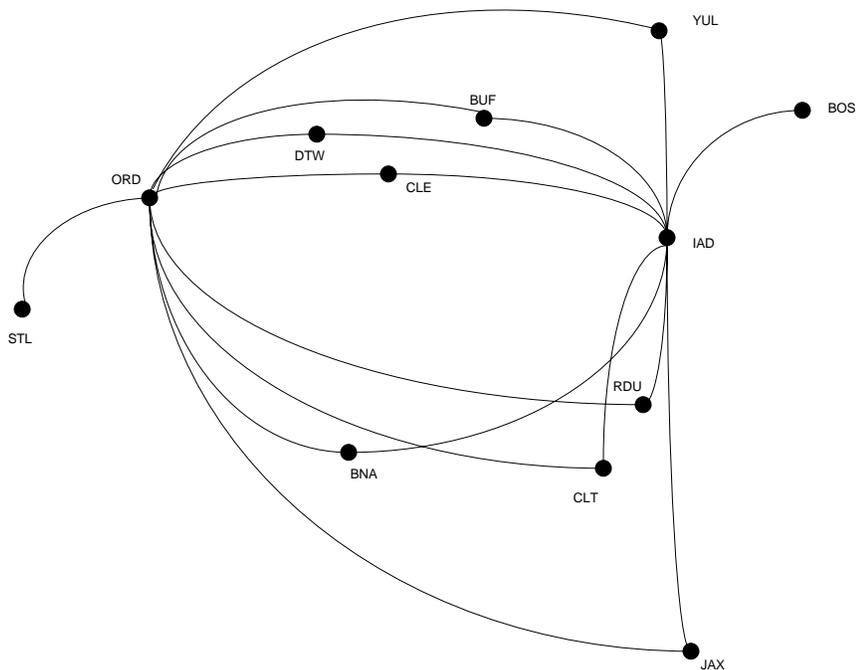


Figure 2.1: A sample airline network with two hubs (ORD and IAD) and nine spokes (Used with permission of Ashgate, 2009; Source: *Airline Operations and Scheduling*, Bazargan, 2004).

2.2 Fleet Assignment

Following the construction of a flight schedule and its corresponding flight network, a fleet assignment problem is solved to determine the type of aircraft for serving each flight in a given schedule, while satisfying operational requirements, maintenance rules, and crew restrictions. In this stage of the planning process, only aircraft types are considered, as opposed to each individual aircraft. A fleet assignment problem was first modeled by Hane et al. (1995), who solved a problem having up to 11 fleets and 2,500 flight legs. This model was simplified by the assumption that each flight is flown every day of the week, which facilitated a reasonable solution of practically sized problems. As a sequel to this work, Clark et al. (1996) generalized the basic fleet assignment model by including maintenance and crew scheduling considerations while preserving solvability. Solutions that violated maintenance constraints were declared subject to being modified in a post-optimization phase, but resulted in a large penalty in the model's objective function.

The *time-space network* structure facilitates the assignment of aircraft to flights by superimposing a set of networks, one for each fleet type having nodes representing points in time or events such as arrivals and departures, along with ground arcs, flight arcs, and wrap-around arcs (see Figure 2.2). The time-space feature arises because the nodes represent the arrival or departure of a flight in a specific airport/city at a given time. Berge and Hopperstad (1993) and Hane et al. (1995) used this time-space network structure representation, which has now become the primary framework for formulating the fleet assignment problem. However, as Rushmeier and Kontogiorgis (1997) point out, the time-space network is not able to distinguish among the specific aircraft on the ground, which limits its application in the subsequent aircraft routing problem.

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 operational requirements and the coverage of flights. They used different solu-

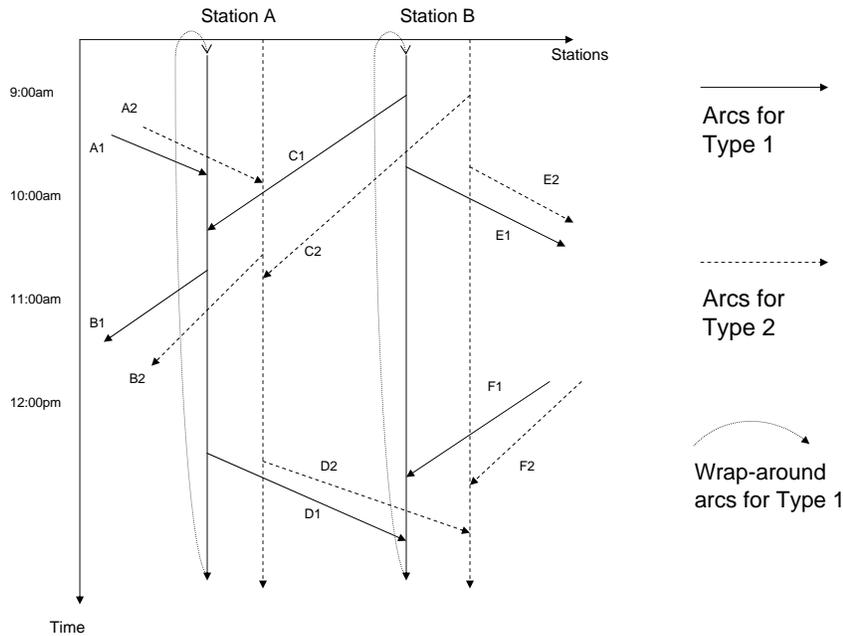


Figure 2.2: A two-type, two-station time-space network.

tion techniques depending on whether there exists a valid initial assignment or not. Flight connection possibilities were also considered, but in order to keep the networks manageably sized, the set of operations at each station was partitioned into connecting complexes representing balanced subsets of the incoming and outgoing flight legs. In addition, Abara (1989) limited the number of possible feasible connections for each flight leg to a small fixed number in the model.

Lohatepanont and Barnhart (2004) dealt with the large problem size and complexity by employing incremental approaches to schedule design, working with a subset of candidate flight legs from a given flight schedule at each step. Kim and Barnhart (2005) considered the problem of designing the flight schedule for a charter airline, developing exact and heuristic sequential solution approaches, and compared their results using data from a charter airline. This kind of problem, also known as the charter airline service network design problem, has

special structures that are imparted by several operating constraints. For example, demands on a given origin-destination (OD) pair are symmetric; at given locations, a minimum number of aircraft of each type are required to overnight; and each aircraft, depending on its type, origin, and destination, is restricted in the number of hours flown daily.

Berge and Hopperstad (1993) proposed a demand driven refueling approach, which deals with the dynamic re-assignment of aircraft to legs within each family in order to respond to rapid market changes, while utilizing improved demand forecasts as departures approach. The refueling decisions were revised frequently prior to flight departures, based on updated demand forecasts using actual bookings.

Fleet assignment decisions need to be made well in advance of departures due to Federal Aviation Administration (FAA) regulations and union contracts, although demand is highly uncertain at this point. To provide flexibility in supply management, airlines have used a sequential framework involving an initial fleet assignment, a refueling process, and finally, certain limited swapping procedures. In the same vein, Sherali et al. (2005) proposed a mixed-integer programming demand-driven refueling (DDR) model considering path-level demands along with a polyhedral analysis to enhance problem solvability. In the DDR problem, the aircraft reassignment is limited within a single family of aircraft and to the flights assigned to the same family as its initial aircraft, thereby still serviceable by the same crew. The advantage of this approach lies in the availability of a more accurate and detailed demand forecast as departure times approach.

Furthermore, Sherali and Zhu (2007) introduced a two-stage stochastic approach where the initial fleet assignment is done first, recognizing that the subsequent refueling process will deal with the stochastic demand fluctuations. To reduce the effect of demand uncertainty, the first stage focuses on making only a family-level fleet assignment to each flight leg, hence providing more flexibility to the subsequent refueling process. Then at the second stage, with improved demand forecasts, a fleet-type level assignment is made for each leg within the same family based on the corresponding path/itinerary demands, without disturbing crew

schedules at hand.

For the sake of illustration, we present below a fleet assignment model based on a time-space network structure, as proposed by Hane et al. (1995):

Notation:

C : set of stations in the network, indexed by o , or d .

F : set of fleet types, indexed by f .

L : set of flight legs scheduled, indexed by i or $\{odt\}$, where $o, d \in C$ and t denotes the time of a takeoff or landing.

H : set of through-flights, indexed by (i, j) , $i, j \in L$.

N : set of nodes in the network, indexed by $\{fot\}$ at time t , where $f \in F$, $o \in C$.

$O(f)$: set of arcs for fleet type f that cross the counting time-line, $f \in F$.

c_{fi} : cost of assigning fleet type f to leg i , $f \in F$, $i \in L$.

$S(f)$: number of available aircraft for fleet type f , $f \in F$.

$$x_{fi} = \begin{cases} 1, & \text{if fleet type } f \text{ covers leg } i, f \in F, i \in L \\ 0, & \text{otherwise.} \end{cases}$$

$y_{fot_1t_2}$: flow of aircraft on the ground arc from node $\{fot_1\}$ to node $\{fot_2\}$ at station $o \in C$ for fleet type $f \in F$.

t^-, t^+ : the point of time preceding and succeeding time t , respectively.

$$\text{Minimize } \sum_{i \in L} \sum_{f \in F} c_{fi} x_{fi} \quad (2.2.1)$$

subject to

$$\sum_{f \in F} x_{fi} = 1, \quad \forall i \in L \quad (2.2.2)$$

$$\sum_{d \in C} x_{fdot} + y_{fot-t} - \sum_{d \in C} x_{fodt} - y_{fott+} = 0, \quad \forall \{fot\} \in N \quad (2.2.3)$$

$$x_{fi} - x_{fj} = 0, \quad \forall (i, j) \in H, f \in F \quad (2.2.4)$$

$$\sum_{i \in O(f)} x_{fi} + \sum_{o \in C} y_{fot_n t_1} \leq S(f), \quad \forall f \in F \quad (2.2.5)$$

$$x \text{ binary}, \quad y \geq 0. \quad (2.2.6)$$

The objective function (2.2.1) minimizes the total cost of assigning fleet types to flight legs. Constraint (2.2.2) ensures that each leg is covered by exactly one fleet type, while Constraint (2.2.3) ensures flow balance at any given node in the network. Constraint (2.2.4) guarantees that if one of two legs of through-flights is assigned to a particular fleet type, then the other leg is also assigned to the same fleet type. Constraint (2.2.5) imposes a limit on the number of available aircraft using a counting time-line that overlaps the wrap-around arcs, in which t_n marks the last event time of the day and t_1 marks the first event time of the day. Finally, the variables representing the flows on the flight arcs and ground arcs in the network are required to be binary and nonnegative, respectively, in Constraint (2.2.6).

Hane et al. (1995) introduced several preprocessing steps such as node aggregation, islands, and the elimination of missed connection, resulting in reductions of the size of rows by a factor of three to six and the size of columns by one to three. They also applied various algorithmic strategies such as cost perturbation, dual simplex with steepest-edge pricing, interior-point method, and branching on cover constraints by prioritizing variables. Particularly, for two large test instances, the variable fixing heuristics significantly decreased the computational times from 1332 and 1540 seconds to 9.7 and 9.4 seconds, respectively.

2.3 Aircraft Routing

The solution obtained from the fleet assignment problem indicates the flow of aircraft types through the flight network. Aircraft routing is the process of identifying which individual aircraft from within each fleet (or aircraft type) is actually assigned to each flight leg. The aircraft routing problem has been addressed by Daskin and Panayotopoulos (1989), Kabbani and Patty (1992), Talluri (1996, 1998), Clarke et al. (1997), Gopalan and Talluri (1998b), and Bartholomew-Biggs et al. (2003). Most solution methods are mainly heuristic in nature. For example, Kabbani and Patty (1992) modeled the aircraft routing problem as a set partitioning problem, where the columns represent week-long routings, while ignoring maintenance constraints. Also, Clarke et al. (1997) presented a flight-based model and described a Lagrangian relaxation solution approach that adds subtour-elimination and maintenance constraints when violated. Relatively more recently, Bartholomew-Biggs et al. (2003) studied the aircraft routing problem related to finding an optimal flight path traversing a minimal distance between a given origin and destination pair while avoiding obstacles in a geographical sense. Direct search, deterministic, and stochastic optimization methods were designed, and related computational results were compared.

Figure 2.3 presents a two-day cyclic routing, where an aircraft starts at one station, and after two days, is routed back to the same station, hence creating a cycle or pattern. Here, one aircraft of type B737-800 starts off at LAX, going through JFK and ORD in sequence, and stays at the JFK for the first night. On the next day, it starts from JFK, going through IAD and JFK in sequence, and ends up at the same station, LAX.

Cohn and Barnhart (2003) incorporated maintenance routing decisions within the crew scheduling problem using a string-based approach. The aircraft routing model is used as a part of this formulation while considering maintenance schedules. For the sake of illustration, we present some details of this string-based assignment model below:

| | Flight No. | Origin | Departure Time | Destination | Arrival Time |
|-------|------------|--------|----------------|-------------|--------------|
| DAY 1 | 101 | LAX | 5:00 | JFK | 13:30 |
| | 129 | JFK | 15:05 | ORD | 16:05 |
| | 109 | ORD | 17:10 | JFK | 20:10 |
| DAY 2 | 140 | JFK | 6:20 | IAD | 7:20 |
| | 120 | IAD | 14:25 | JFK | 15:25 |
| | 127 | JFK | 19:00 | LAX | 21:30 |

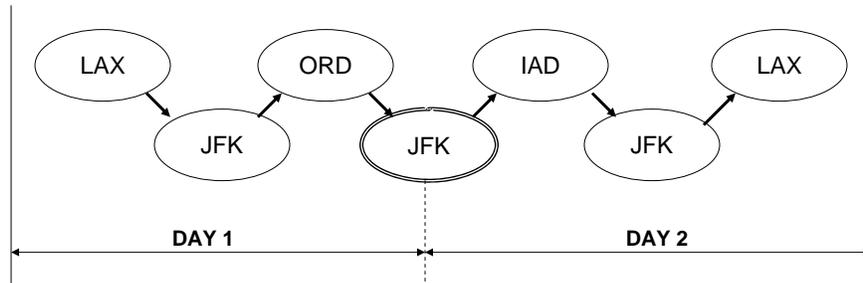


Figure 2.3: Two-day routing for aircraft type B737-800 (Used with permission of Ashgate, 2009; Source: *Airline Operations and Scheduling*, Bazargan, 2004)

Notation:

N : set of nodes, which represent points in space and time at which route strings begin or end (and thus, aircraft are needed or become available).

F : set of flights.

R : set of feasible route strings.

c_r : the cost of route string $r \in R$.

$$\alpha_{fr} : \begin{cases} 1, \text{ if route string } r \text{ contains flight } f, \forall r \in R, f \in F \\ 0, \text{ otherwise.} \end{cases}$$

$$d_r : \begin{cases} 1, \text{ if route string } r \text{ is included in the solution, } \forall r \in R \\ 0, \text{ otherwise.} \end{cases}$$

g_n^-, g_n^+ : ground arc variables representing the number of aircraft on the ground at station s immediately prior to, and immediately following, time t , respectively, given a node n that represents time t at station s , $\forall n \in N$.

R^T : set of route strings that span time T , an arbitrary time known as the *countline*.

N^T : set of nodes with corresponding ground arcs g^+ spanning the countline.

K : available number of aircraft.

$$\text{Minimize } \sum_{r \in R} c_r d_r \quad (2.3.1)$$

subject to

$$\sum_{r \in R} \alpha_{fr} d_r = 1, \quad \forall f \in F, \quad (2.3.2)$$

$$\sum_{r \text{ ends at node } n} d_r + g_n^- - \sum_{r \text{ starts at node } n} d_r - g_n^+ = 0, \quad \forall n \in N, \quad (2.3.3)$$

$$\sum_{r \in R^T} d_r + \sum_{n \in Z^T} g_n^+ \leq K, \quad (2.3.4)$$

$$d_r \in \{0, 1\}, \quad \forall r \in R, \quad (2.3.5)$$

$$g_n^-, g_n^+ \geq 0, \quad \forall n \in N. \quad (2.3.6)$$

The objective function (2.3.1) minimizes the total cost of the route strings. The first constraint (2.3.2) is a *coverage constraint*, which asserts that each flight must be included in exactly one chosen route string. The second constraint (2.3.3) is a *balance constraint*, i.e., for any given node n , the number of aircraft on route strings ending at n plus the number of aircraft on the ground arc entering into n must equal the number of aircraft on route strings starting at n plus the number of aircraft on the ground arc leaving node n . Constraint (2.3.4) requires that the total number of aircraft at the countline, T , does not exceed the total number of available aircraft, K . The route string variables are declared to be binary-valued in Constraint (2.3.5), and the integrality of the ground arc variables is relaxed in Constraint (2.3.6).

Cohn and Barnhart (2003) used a branch-and-price solution approach to solve the crew pairing problem and extended this methodology to incorporate maintenance routing problems as well. More specifically, starting with an initial set of crew pairings and maintenance solutions at the root node, they generated both a negative reduced cost crew pairing and a negative reduced cost maintenance solution by utilizing the corresponding dual variable values after solving the LP relaxation, and then updated lower and upper bounds on the overall model, if necessary, during the iterative process. For a small number (16 to 20) of given maintenance solution columns, all feasible pairings were enumerated, and the authors were thus able to generate solutions having at most a 2% optimality gap.

2.4 Crew Scheduling

The crew scheduling process involves identifying sequences of flight legs and assigning crews to them and, similar to aircraft routing, is usually done after the fleet assignment has been performed. Crew costs constitute a large portion of the airline expenses, and are second only to fuel costs. However, unlike fuel costs, crew costs are mostly controllable from the management point of view, that is, savings in crew costs can be achieved through better scheduling. A *crew pairing* is a sequence of flight legs flown by the same aircraft type, starting and ending at the same station, called the crew base. Crew pairings need to satisfy constraints such as union contracts and FAA regulations, and the objective of crew scheduling is to find a set of pairings to minimize the total crew cost of assigning crews to flight legs such that every flight is covered. Typically a pairing is composed of several duty periods with overnight rests between them, where a *duty period* is a working day for a crew that consists of several flight legs with a *sit connection* (short rest period) between them. Crew scheduling problems are usually formulated as set partitioning models where the columns represent feasible crew pairings and the rows represent the scheduled flights. Hence, since such a model involves a huge number of variables, the main proposed solution approaches are

based on column generation techniques (e.g., see Minoux (1984), and Lavoie et al. (1988)), which are then combined with branch-and-price procedures (Ribeiro et al. (1989), Gamache et al. (1994), Desaulniers et al. (1997a), Vance et al. (1997a), Barnhart and Shenoi (1998), Barnhart et al. (1998b), Gamache et al. (1998), Stojkovic et al. (1998), Lettovsky et al. (2000), and Yan and Chang (2002)). A branch-and-cut method was also proposed by Hoffman and Padberg (1993) and a Lagrangian relaxation based heuristic was adopted by Alefragis et al. (2000).

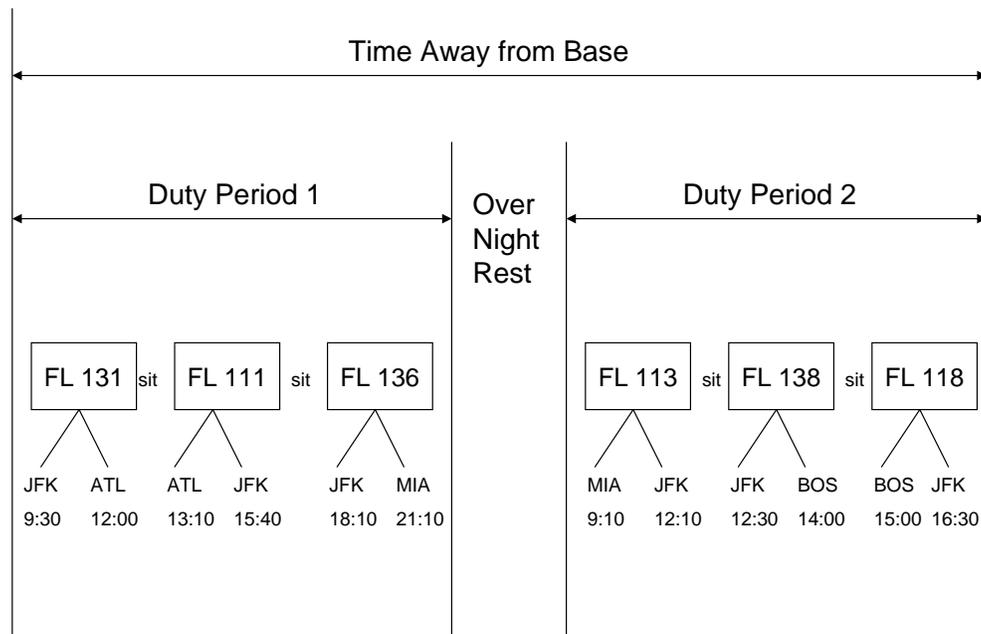


Figure 2.4: A pairing with duty periods, sits, and overnight rests (Used with permission of Ashgate, 2009; Source: *Airline Operations and Scheduling*, Bazargan, 2004)

Figure 2.4 illustrates a two-day crew pairing, which forms a sequence of duties separated by an overnight rest period, and where each duty period consists of flight legs and sits between

them. This two-day pairing shows that the crew is based at JFK, serves three flights (JFK to ATL, ATL to JFK, and JFK to MIA), and stays overnight away from the home-base on the first day. On the following day, the crew starts from MIA, serves three flights (MIA to JFK, JFK to BOS, BOS to JFK), thus coming back to the base, JFK.

Barnhart and Shenoi (1998) developed an approximate model for the crew pairing problem by relaxing time-away-from-base constraints and restrictions on the start and end locations for pairings. Hence, by replacing pairing variables with duty variables, the crew pairing problem was formulated as a network flow model, as detailed below for clarifying concepts.

Notation:

n : number of duties.

m : number of operational flights.

p : number of ground arcs.

q : number of network nodes.

$$x_j : \begin{cases} 1, \text{ if duty } j \text{ is flown} \\ 0, \text{ otherwise.} \end{cases}$$

y_j : number of crews using ground arc j .

c_j : elapsed time cost of duty j plus the costs of deadheads in duty j .

d_j : elapsed time cost of ground arc j .

$$a_{ij} : \begin{cases} 1, \text{ if operational flight } i \text{ is included in duty } j \\ 0, \text{ otherwise.} \end{cases}$$

$$b_{lj} : \begin{cases} 1, \text{ if duty } j \text{ or ground arc } j \text{ enters node } l \\ -1, \text{ if it leaves node } l \\ 0, \text{ otherwise.} \end{cases}$$

$$\text{Minimize } \sum_{j=1}^n c_j x_j + \sum_{j=1}^p d_j y_j, \quad (2.4.1)$$

subject to

$$\sum_{j=1}^n a_{ij} x_j = 1, \quad \forall i = 1, \dots, m, \quad (2.4.2)$$

$$\sum_{j=1}^n b_{lj} x_j + \sum_{j=1}^p b_{lj} y_j = 0, \quad \forall l = 1, \dots, q, \quad (2.4.3)$$

$$x_j \in \{0, 1\}, \quad \forall j = 1, \dots, n, \quad (2.4.4)$$

$$y_j \in Z^+, \quad \forall j = 1, \dots, n. \quad (2.4.5)$$

The objective function (2.4.1) is to minimize the cost of selected duty periods and the cost of elapsed time on ground arcs. Constraint (2.4.2) ensures that each flight is covered exactly once by some selected duty j , and Constraint (2.4.3) requires the conservation of flow of crews at each node l . The duty variables are declared to be binary-valued in Constraint (2.4.4), and Constraint (2.4.5) requires integrality for the number of crews on each ground arc j .

Barnhart and Shenoi (1998) first solved an approximate model for the crew pairing problem by relaxing some restrictions on pairings and judiciously selecting deadheads to improve the solutions. Starting with these solutions, a restricted column generation procedure was used to facilitate the solution process. The authors reported optimality gaps within 0.85% on average, and a reduction of 20% in solution costs and a reduction of 80% in run-times.

2.5 Integrated Models for the Fleet Assignment Problem (FAP)

Since the different airline operations planning processes are related to each other, several integrated models have been recently proposed that simultaneously consider combinations

of these problems in order to find a better solution for the entire system. Particularly, an integrated fleet assignment and schedule design model can increase revenues by allowing for improved flight connection opportunities in the FAP. Desaulniers et al. (1997) and Rexing et al. (2000) assumed that the flight origin-destination information is given, but the flight departure times are allowed to vary. By discretizing the set of departure times for each leg within its specified time-window, they provided more choices for assigning aircraft to flights, thus improving the fleet assignment solution.

Desaulniers et al. (1997) proposed integrated routing and fleet assignment models by pre-specifying sequences of flights and assigning these sequences to fleet types, and used branch-and-price schemes with departure time-windows imposed within the subproblems. Similarly, Barnhart et al. (1998) introduced a string-based model to solve the weekly fleet assignment and aircraft routing problems simultaneously. They defined a string as a sequence of connected flights that begin and end at maintenance stations and explicitly accommodated maintenance considerations using ground arcs. In addition, this model can be used to determine through-flights, since a string corresponds to a sequence of legs flown by a single aircraft. On the other hand, Ahuja et al. (2002) developed a Very Large-Scale Neighborhood Search Algorithm, which starts from a feasible fleet assignment solution and then determines the through-flights sequentially.

The integrated model by Yan and Tseng (2002) combines the scheduling process with fleet assignment, while including itinerary/path-based demand considerations. In this model, all the flight legs are considered as optional legs and are selected as needed to include into the schedule, hence constructing passenger paths. Subsequently, Yan et al. (2005) presented an integrated scheduling and routing model while incorporating passenger choice behavior, which was formulated as a nonlinear mixed-integer program and solved using a heuristic.

Klabjan et al. (2002) devised an integrated approach for schedule design, aircraft routing, and crew scheduling by changing the traditional solution sequence, i.e., they solved the crew scheduling problem before the routing problem, and added plane-count constraints to the

crew scheduling problem to maintain the feasibility of aircraft routing.

Cordeau et al. (2001) proposed a solution approach based on Benders' decomposition where the aircraft routing problem is solved at the master program level, and the crew pairing problem at the subproblem level. Similarly, but conversely, Mercier et al. (2005) suggested an improved approach in which the sequence is reversed, i.e., the crew pairing problem is solved at the master program level and the aircraft routing problem is solved at the subproblem level.

Cohn and Barnhart (2003) developed a model for the integrated crew scheduling and aircraft routing problem that additionally incorporates certain key maintenance routing decisions. Due to the large number of variables, the authors suggested generating only a subset of columns in a preprocessing step by solving a series of aircraft routing problems.

Sandhu and Klabjan (2007) proposed a model for simultaneously considering fleet assignment, aircraft routing, and crew scheduling, where the aircraft routing problem was approximately accommodated by utilizing certain suitable plane-count constraints. The proposed integrated model was solved using Lagrangian relaxation combined with column generation, and also via a Benders' decomposition procedure, where the former methodology was demonstrated to perform better. In order to control tractability, the fleet assignment model was solved over all the aircraft families and flights first, and then a particular subset of aircraft families and the corresponding flights were selected to be considered in the integrated model.

Mercier and Soumis (2007) proposed an integrated aircraft routing, crew scheduling, and flight retiming model, assuming that the fleet assignment problem has been solved as a preprocessor, and hence, the aircraft type assigned to each flight leg is known. Accordingly, given a set of flight legs to be flown by each aircraft of a specific type, the problem is to determine a modified schedule and a set of aircraft routes and crew pairings that minimize the cost of operation for each aircraft type. Benders decomposition combined with column generation and a dynamic constraint generation procedure was used to solve this model.

Leg-based fleet assignment model solutions do not always reflect the actual revenue properly, especially for larger aircraft, since more multi-leg passengers are likely to fly via such larger aircraft-based flights. Kniker (1998), Jacobs et al. (1999), and Barnhart et al. (2002b) considered origin-destination fleet assignment models where passenger revenues are modeled for each origin-destination itinerary by using a passenger-mix model to decide on the booking level of the number of passengers in each fixed-seat capacity aircraft. However, there exists no published work on integrated models that factor in revenue management.

2.6 Schedule Disruption and Recovery

In the previous sections, we have discussed some of the airline scheduling models that arise at different planning stages, which tightly manage resources such as aircraft and crews. However, in practice, the planned flight schedules are often disrupted during the days of operations due to several reasons such as inclement weather, mechanical problems, unavailable crews, airspace congestions, and ground holds. Generally, these disruptions affect the execution of regular operations, and sometimes, the deviations from the original flight plans are large enough to impose substantial changes on the network. Furthermore, each flight delay can propagate to disrupt subsequent downstream flights that await aircraft and crew of the delayed flights, and a small disruption could result in a huge disrupted leg set throughout the network. These disruptions are compounded due to the fact that the resources can split. For example, an aircraft and crew might be assigned to the same flight at a particular point in the schedule, but would be reassigned to separate flights at a later point, while the passengers are on itineraries where some of the legs do not involve any of the previously assigned aircraft or crews.

Robustness can be seen as a proactive way of handling disruptions by having flight schedules, aircraft routings, or crew rotations less sensitive to situational changes, while keeping the schedules feasible or by facilitating different types of recovery actions. For instance,

slack time can be built into the schedule to absorb disruptions, standby reserve crews can be strategically located at certain stations, and aircraft or crew schedules can be partitioned into sections in order to prevent delays occurring in one section from spreading to the remaining flight network. Also, as a preemptive measure to increase the schedule robustness, the flight schedule network can be designed such that many short cycles in the aircraft rotations are included. However, there is a trade-off between robustness and cost, e.g., imposing short cycle restrictions on aircraft rotations would not be able to achieve the same level of optimality as otherwise. In addition, there are two naturally opposing viewpoints regarding slacks in schedules; slack is typically undesirable from the planning perspective because it consumes costly resources, whereas it may be critical from an operational perspective for absorbing disruptions and preventing their subsequent propagation.

While robust planning models try to avoid disruptions in advance, recovery models seek the best way of responding to the aftermath of disruptions, minimizing their impact on the system and preventing further propagation. In this context, four different types of disruptions can occur: (1) a set of delayed flights throughout the network; (2) unscheduled maintenance; (3) unavailability of crews; and (4) airport capacity reduction or airport closure. The common recovery strategies are delaying flight departures, canceling flights, or swapping aircraft and crews, where these measures can be interrelated. For example, in recovering the disrupted crew pairing, it might be necessary to delay or cancel more flights or swap with another crew pairing, causing a disruption in additional aircraft rotations. This chained effect significantly complicates the problem. Hence, while defining recovery strategies for schedule changes, aircraft rerouting, and crew rescheduling, it is critical to determine a proper recovery scope and to design a reasonable recovery plan in order to assure that the formulated problem is computationally tractable in real-time.

The real-time nature of (tactical) recovery problems demands relatively short computational solution times as compared with the (strategic) operational planning problems. One way to achieve this is to reduce the size of the related models. The disrupted part of the original flight schedule is thus limited by a time-window, which can span from the beginning

time of disruption up to a certain number of hours, or until the rest of the day of operation. Likewise, the number of crew members included within the recovery problem is also limited by only including the affected crew members and possible reserve crew members.

Teodorovic and Stojkovic (1995) designed an approach for determining a new flight schedule and aircraft rotations in case of disruptions. In this method, legs are grouped together within the crew rotations and crew rotations are grouped together within the aircraft rotations. A heuristic is used to generate the crew rotations, and possible deadheadings are considered in case that crews do not automatically return to their crew bases in the planned schedule.

Jarrah et al. (1993) presented two network flow models for solving the aircraft recovery problem, and used flight delays, cancellations, aircraft swappings, and reserve aircraft as recovery strategies. In their approach, flight cancellations were not allowed for one model whereas delaying flights was not considered in the other model, and the authors suggested the possibility of finding better solutions by combining delays and cancellations to capture their interactions. In order to assess the cost of delaying or cancelling a flight, the authors constructed a disutility function, based on the total number of passengers, the number of passengers with a downstream connection, lost crew time, and disruption of maintenance.

Cao and Kanafani (1997) presented a model where the objective is to maximize flight revenues minus swapping and delaying costs. Both delays and cancellations, ferrying (i.e., flying a deadhead aircraft to a station for the next operation), and multiple aircraft type swappings were taken into consideration in their model.

Clarke (1997) proposed a comprehensive framework for reassigning operational aircraft to scheduled flights in the aftermath of irregularities. Multiple aircraft type swapping, flight delays and cancellations, as well as the impact of air traffic management initiatives and crew availability were incorporated in the model.

Thengvall et al. (2000) used a multi-commodity network flow model for recovery following a hub closure. Using the framework based on a time-space network with flight arcs, ground arcs, and overnight arcs, the authors added arcs for ferrying, arcs for time-shifted copies of

original flights (accommodating delays), protection arcs, and through-flight arcs. This model can handle delays, cancellations, as well as swaps among different fleets. The authors solved the problem with various objective functions such as minimizing total cancellation and delay costs, and preserving as many of the original aircraft routings as possible.

One proactive approach to address disruptions is to generate airline schedules that can isolate the impact of disruptions such that any delay or cancellation does not impact the operations throughout the airline network. In this spirit, Kang (2004) proposed a methodology for deriving robust airline schedules, or *degradable schedules*, where the existing schedule is partitioned into several independent sub-schedules or layers, and each layer has a different level of importance or priority. The basic logic behind a degradable airline schedule is that it can simultaneously increase the robustness of an airline's operation, since the aircraft routings within each layer are independent of those in other layers, and therefore, the impact of a delay or cancellation in a layer is limited within that layer and does not propagate to other layers. Another benefit of a degradable airline schedule is that it can segment the products offered to passengers based on their reliability, since the partitioning of the schedule serves to assign different reliabilities to itineraries. If layers are prioritized in the order that delays and cancellations are assigned, the reverse order of priority can function as an indicator of reliability of the itineraries within each layer. The author presented three different integer programming models that incorporate degradability into the scheduling process: Degradable Schedule Partition Model (D-SPM), Degradable Fleet Assignment Model (D-FAM), and Degradable Aircraft Routing Model (D-ARM), which are respectively designed to be applied between the schedule design and fleet assignment steps, to be solved along with the fleet assignment problem, and to be solved along with the aircraft routing problem. The results showed that flight legs within high revenue itineraries can be assigned to a higher priority layer without changing the original schedule or using additional aircraft, and also that it is possible to discriminate among the different levels of service quality for passengers within different priority layers.

Nissen and Haase (2006) proposed a duty-based formulation for the crew recovery problem.

This modeling approach is especially well suited for solving the crew disruption problem when taking into account fixed monthly crew rates. The disruption is resolved within each duty period, hence requiring shorter recovery horizons and thus leading to a reduction in the problem size. The authors applied a branch-and-price-based solution method using a set covering formulation within the master problem, and a resource-constrained shortest path formulation as the subproblem.

Lan et al. (2006) considered how changes in aircraft routings can be used to reduce the potential delays that propagate over connecting aircraft. In this case, flight departure times were held unchanged, but a reassignment of aircraft to flights was allowed to achieve a better utilization of the slack in the system for absorbing disruptions. In a separate case, flight times were allowed to vary within a limited time-window while keeping the aircraft routings fixed. The objective used in this problem is to minimize the delay impact on passengers making flight connections.

AhmadBeygi et al. (2010) developed an approach that can improve the operational performance of a planned airline schedule without increasing its planned cost. This was done by retiming flight departure times in order to redistribute the existing slack to those flight connections where it can be best utilized. Furthermore, restrictions were imposed on flight retimings such that the projected demand does not change, the crew pairings remain feasible, and the same aircraft rotations can be maintained.

Recovering passenger itineraries is important to airlines since this process involves additional operational costs related to passenger delays, as well as possible passenger dissatisfaction leading to a loss of goodwill. Bratu and Barnhart (2006) proposed two passenger recovery models, a Passenger Delay Metric (PDM) model and a Disrupted Passenger Metric (DPM) model, where flights can be delayed or cancelled, and where reserve crews and aircraft are assigned to the flight legs. In their model, every flight is represented by several arcs, one for each departure time, allowing flight departures to be retimed. While some regulations for the reserve crew are incorporated within the models, recovery of the disrupted

crew is not taken into consideration. In the PDM model, delay costs are explicitly modeled by considering the costs incurred by disruptions and recovery options, whereas delay costs are only approximated in the DPM model.

The typical hierarchical structure of disruption recovery in practice is that the flight schedule is recovered first, and the crew and aircraft re-scheduling decisions are made next, and finally, passengers are recovered. However, in this sequential recovery process, when taking into account crew recovery, the flight delay or cancellation decisions might result in crew pairings becoming infeasible. The benefit of integrating the recovery of these several resources lies in simultaneously handling aircraft and crews, and only a few such attempts to solve integrated recovery problems have been reported in the literature.

The differences between integrated recovery models and integrated planning models are that in the former: *(i)* schedule adjustments need to be modeled due to various delay or cancellation options for the legs involved; *(ii)* the flight legs within the recovery scope need to be dated (instead of being daily events); and *(iii)* crew recovery is specific to individual crew members rather than to a crew base. Since the time-space network for fleet assignment also becomes dated, there is no wrap-around (circular) arc that is necessary to connect the beginning and the end of a day. At the end of the recovery time-window, the number of aircraft of any particular aircraft type at each station is expected to be the same as that in the original schedule.

Lettovsky (1997) introduced an integrated recovery framework and presented an MIP model that maximizes total profits while capturing the availability of the resources of aircraft, crews, and passengers by defining four sets of decision variables for: *(i)* assigning a certain aircraft type to the flight leg; *(ii)* assigning a certain pairing to the crew; *(iii)* rerouting the aircraft; and *(iv)* rescheduling the flight legs. The formulation has three parts corresponding to each of the resources, i.e., crew assignment, aircraft routing, and passenger flow. Benders decomposition was used as a solution approach, where the Schedule Recovery Model (SRM) constituted the master program, which determines a plan for cancellations, delays, and fleet

assignment. Following this, in the subproblems, the Aircraft Recovery Model (ARM_f) for each aircraft type f and the Crew Recovery Model for each crew group c (CRM_c) were solved. The Passenger Flow Model (PFM) was finally used to evaluate the passenger flows.

Gao (2007) proposed an integrated recovery model that integrates the schedule recovery, fleet reassignment, aircraft rerouting, and crew rescheduling processes. The schedule adjustment step was incorporated within a dated fleet assignment (FAM) and crew scheduling model, where crew rescheduling was represented by using a duty partition approach. The author applied heuristics for selecting flight delay options, move-up crews, and potential aircraft swappings, and used Benders decomposition as a solution approach, where the duty flow model was combined with FAM in the master program, while the aircraft routing decisions were made in the subproblem.

2.7 Revenue/Yield Management (RM/YM)

According to Smith et al. (1992), from the early 1960's, American Airlines began research in managing revenue from its reservations inventory. With the emergence of SABRE (Semi-Automated Business Research Environment), a subsidiary of American Airlines, yield management was introduced to control and manage the reservations inventory in a way that increases profits, given the flight schedule and fare structure. Once the combination of schedule and fare defines the products to be offered to the public, yield management then determines how much of each product to put on the shelf, making it available for sale. The yield management system of American Airlines is divided into three major functions: overbooking, discount allocation, and traffic management. In *overbooking*, four forecasts are used: (1) probability of a cancellation; (2) probability of a no-show; (3) recapture probability; and (4) oversale cost. In *discount seat allocation*, a passenger may request a discounted fare, and it will be either accepted or rejected. If it is rejected, then the passenger may choose to pay a full fare or turn away. Seats are partitioned into several groups of fare

types, mostly according to their values. To protect their allotments in each fare type, hence increasing profits, nested fare-classes are used to ensure that the lowest-valued fare-classes close first and that the full fare-class is closed only if the flight reaches its overbooking level. In *traffic management*, where multi-leg flights with connections are considered, American Airlines developed a method of clustering fare-classes into a small number of similarly valued groupings called *buckets*. The buckets are nested so that as sales increase, availability is restricted first to higher value reservations. With regard to nesting in traffic management, fare-class availability is determined for each request by combining indexing with bucket availability, meaning that a fare-class is available for sale only if its corresponding bucket has seats available. American Airlines' Decision Technologies developed a dynamic programming algorithm to index fare-classes into buckets. The objective is to minimize the variability of fare-class values within buckets and to maximize the variability between buckets. The revenue performance of overbooking or discount allocation is measured by comparing the actual net revenues to the estimated revenues that could have been earned with perfect controls. This method is referred to as measuring the revenue opportunity. Reliable recapture estimations can be difficult using traditional statistical methods because of the sometimes small effects being measured in a very dynamic environment. It is mentioned that recapture has been accurately measured using passenger choice modeling. However, no explicit model is specified for this purpose in Smith et al. (1992).

As Delta Airlines flies over 2500 domestic flight legs every day with 450 aircraft from 10 different fleets, the "Coldstart" model was introduced by Subramanian et al. (1994) to enhance the fleet assignment problem. This model is a large-scale mixed-integer linear program that assigns fleet types to flight legs in order to minimize a combination of operating and passenger spill costs, subject to a variety of operational constraints within a single day time-window. It assigns fleet types to the flight legs, and then individual aircraft are routed after the model is solved to ensure that the solution is feasible. Three objective functions are used in the "Coldstart" model: (a) minimizing the cost; (b) minimizing the total number of planes used to fly the schedule; and (c) maximizing the profit. The model typically has

40,000 constraints and 60,000 variables (20,000 binary variables and 40,000 general integer variables). The fleet assignment model is built on a time-space network as a fundamental mathematical structure. The interior point code, OBI, is first used to solve the continuous linear programming (LP) relaxation in order to reduce the size of the problem by fixing some or all of the binary variables, and then the mixed-integer programming code, OSL, is used to solve the resulting smaller problem. Methods used in the “Coldstart” model in this paper are now outdated, and have since been replaced or updated. However, the paper by Subramanian et al. (1994) provides the basic concepts of fleet assignment modeling while considering various aspects using simple mathematical formulations and examples. The authors briefly address the recapture issue in regard to spill cost, which is estimated based on the demand for a leg, the aircraft capacity, the recapture rate, and the revenue associated with the lost passengers. It is mentioned that the percentage of spilled passengers that are recaptured on other Delta flights on a leg-by-leg basis (recapture rate) is estimated using a market-share model, which is not specified.

In the airline industry, pricing in the presence of high fixed costs and negligible variable costs is addressed by heuristics within the realm of yield management. Production processes, in general, are characterized by high costs for capacity resourcing and have relatively low variable marginal costs for an additional unit of output, which is the cost of serving an additional passenger in the context of airline management. Since each additional demand unit leads to a high contribution margin, an appropriate price and capacity management provides a reasonable potential for profit increase. Schwind and Wendt (2002) argue that pricing many complementary resources cannot be solved by the simple addition of value functions, and so they introduced machine learning techniques, thereby providing for an efficient adaptive pricing of resource attributes related to the multidimensional yield management problem. In this paper, the dynamic pricing of information services for market demands shows high similarities to yield management problems that arise in several service industries, especially in the airline industry. Two different solution methodologies were devised based on a genetic algorithm and reinforcement learning, where the latter was demonstrated to perform better.

To find optimal booking limits for a given flight leg and fare-classes in airline yield management problems, the demand forecast by fare-class and passenger itinerary, and an estimate of the fare value associated with each category of forecasted demand, are required as inputs to the models. Weatherford and Belobaba (2002) focused on these two critical inputs and examined impacts of errors in the demand forecasts and fare estimates on the revenue performance of commonly used seat allocation heuristics. Independent demands in each fare-class, no cancellations of bookings, lost spilled passengers (no recapture), and a complete booking of lower fare-classes before higher fare-classes were assumed, and a simulation analysis was conducted for a single flight leg with multiple booking classes. It was shown that inaccurate forecasts can actually have substantially negative revenue impacts and, furthermore, over-forecasting tends to have a larger negative impact on revenues than under-forecasting in cases where the demand is almost equal to the capacity. In addition, it was discovered that there are some cases where misestimation of the mean demands and mean fare values used as inputs can actually be beneficial, while the variability of actual fare values around the mean fare values does not have a significant impact.

Some of the important aspects related to revenue management problems are ticket pricing, seat or discount allocation, and overbooking. A more realistic model can have multiple fare-classes, overbooking of flights, concurrent demand arrivals of passengers from different fare-classes, and class-dependent but random cancellations. With all these factors, Gosavi et al. (2002) considered the single leg revenue management problem as a continuous-time semi-Markov decision process with random demand arrivals and cancellations, and used reinforcement learning as a tool for solution. It was shown that the nested expected marginal seat revenue model by Belobaba (1989) can be outperformed by using the proposed reinforcement learning model. Recently, more attention has been focused on the network version of the problem in which booking is done on the basis of not just the leg of interest, but on considering paths in the entire network (this is called the *network effect*). Gosavi et al. (2002) claimed that since the fare-class is based on the itinerary spread across the entire network, the network effect is partially accommodated in the model. In addition, the network effect,

to some extent, is incorporated by considering the concurrent arrival model by Robinson (1995), rather than assuming that fare-classes arrive sequentially.

In maximizing operating profits in the airline industry, a simple yield management strategy using a genetic algorithm was proposed and analyzed by Pulugurtha and Nambisan (2003) in order to estimate the number of passengers belonging to each fare-class. The model was tested using various scenarios, which were developed to validate the proposed methodology, while considering overbooking, ticket cancellations, time-varying demand, and the effect of continuing flights. However, this model is simplified in a sense that demand is assumed to be independent across fare-classes, the number of seats sold in each fare-class is restricted, and the same weight is given to passengers on each single segment of continuing flights rather than ascribing more weight to passengers on multiple segments.

Some practical ways to improve airline revenue management were suggested and examined by Parker (2003), such as setting and following standard protocols and procedures to maintain a well-calibrated revenue management system, and sharing knowledge and ideas among ‘the community of practice’, which includes the scheduling, pricing, and operational departments. In addition, the ability to manipulate correctly the number of booking class seats that are protected for each type of customer has become a critical skill for optimizing revenues. In the paper by Parker (2003), flights are classified as having a prime (P), medium (M), or low (L) passenger load-factor and, at the same time, as having a business (B), medium (M), or leisure (L) passenger-mix, which reflects the percentage of high-paying passengers. Hence, with historical passenger load levels and booking class distribution dates, each flight can be identified by a two-letter combined code. For seat protection manipulation, when seat bookings reach a pre-defined level or a pre-defined time prior to departure, ‘sell-up’ manipulation is initiated, which allows the most discounted booking class to be ‘squeezed’, thereby pushing seat availabilities to higher priced seat classes.

A dynamic management capacity strategy was studied by Frank et al. (2005) to evaluate revenue impacts of a continuously adjusted fleet assignment during the booking period.

Historically, the fleet assignment involves an optimal matching of medium-term demand forecasts to available capacity. The motivation is that forecasts usually improve in time since the number of bookings at hand add more information to the possible outcome. With the use of simulation, some useful results regarding the impact of continuous capacity adjustment on revenue can be obtained through sensitivity analyses with respect to parameters such as the demand volume and ‘sell-up’ probability. To determine the influence of a flexible capacity allocation on the achievable revenue, Frank et al. (2005) used simulation by generating demand patterns with time-dependent distributions of bookings, fluctuations in total volume, and interdependencies of demand in different booking classes. After a flight schedule was built based on this demand analysis, aircraft were assigned to the legs according to the latest passenger demand forecasts, while considering restrictions such as the number of available slots and the connection quality over a limited time-window. Within this time-window, aircraft swaps were allowed between inbound and outbound flights. As the authors mention, in order to practically enhance the opportunity to revise capacities and aircraft assignments, it is desirable to have the crew planning process shortened. It is notable that due to the swapping of aircraft, seat capacity stays above the given number of bookings at hand.

Dunleavy and Westermann (2005) discussed future trends in airline revenue management. Historically, airlines have adopted the idea of market segmentation to maximize revenues. Unlike major airlines that impose sets of complex rules and conditions on the fare levels, the low-cost carriers (LCCs) provide different price levels for a single product, depending on how close the day-of-departure is to the day-of-booking. Moreover, the LCCs have been able to acquire modern, fuel-efficient fleets at relatively competitive rates. However, as competition increases in the marketplace, the differential margins will continue to shrink and, although some of the traditional airlines may not survive, the same is true for many of the LCCs. A future pricing mechanism is likely to be constructed considering that there are different customer segments having different demands at the time of purchase. The ability of airlines to differentiate between price-sensitive and product-sensitive consumers will be critical in ascertaining the future role of airline revenue management. Dunleavy and

Westermann (2005) approached the topic of airline revenue management in a very general way, introducing the change of market structure due to the LCCs and considering the internet travel search sites available to customers, without utilizing any model-based methodologies or solutions. Ultimately, revenue management will evolve to the determination of what is the most appropriate fare for a customer at a particular time prior to departure, in order to ensure the highest probability of a sale.

Over the past forty years, airline revenue management has evolved from low level inventory control methods to advanced information systems. In order to increase passenger load factors, and thereby profits, it is necessary to analyze passenger demands and control seat allocations effectively. Chang et al. (2006) proposed a case-based reasoning technique to solve the decision problem of seat allocation and the decision process of accepting booking requests based on booking information. A first-come-first-served protocol, an expected dynamic probability method, and a case-based seat allocation system were used and the results were compared, indicating that the case-based seat allocation system outperforms the other methods. However, this model is restricted and simplified by assumptions such as the booking demand for each fare-class being independent, the influence of other competitors being negligible in the seat allocation planning process, and by the absence of recapture considerations if a booking request is declined.

With this discussion, we now proceed to present the analytical contributions of this dissertation.

Chapter 3

An Integrated Schedule Design and Fleet Assignment Model

3.1 Introduction

In this chapter, we develop and analyze a model that integrates flight scheduling and fleet assignment, while considering optional legs, path/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 (the latter is considered in Chapter 4). 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.

This chapter makes the following specific contributions. First, we present a new mixed-integer programming model that integrates flight scheduling and fleet assignment, while directly incorporating 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 in order 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 US airline carrier, *United Airlines*. Our computational study demonstrates the efficacy of the proposed procedures and reveals a substantial potential increase in profitability.

The remainder of this chapter is organized as follows. In Section 3.2, we present the basic integrated mixed-integer programming model. Section 3.3 introduces several classes of valid inequalities along with accompanying separation routines to generate them for tightening the model representation. In Section 3.4, we discuss two different algorithmic approaches employing Benders decomposition to efficiently optimize the developed model. Finally, results from our computational experiments using real airline data are reported and analyzed in Section 3.5.

3.2 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), and 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:

$$\mathbf{FSFAM1} : \text{Maximize } \sum_{p \in \Pi} \sum_{h \in H_p} f_{ph} \pi_{ph} - \sum_{a \in AT} \sum_{j \in L} c_{aj} x_{aj} \quad (3.2.1)$$

subject to:

$$\sum_{a \in AT} x_{aj} = 1, \quad \forall j \in L^M \quad (3.2.2)$$

$$\sum_{a \in AT} x_{aj} \leq 1, \quad \forall j \in L^O \quad (3.2.3)$$

$$\sum_{j \in L} b f_{jn} x_{aj} + \sum_{g \in G_a} b g_{gn} w_g = 0, \quad \forall n \in N_a, \forall a \in AT \quad (3.2.4)$$

$$\sum_{j \in CS_a} x_{aj} + \sum_{g \in CS_a} w_g \leq N A_a, \quad \forall a \in AT \quad (3.2.5)$$

$$\sum_{p \in \Pi(j)} \pi_{ph} \leq \sum_{a \in AT} C a p_{ah} x_{aj}, \quad \forall j \in L, \forall h \in H \quad (3.2.6)$$

$$\pi_{ph} \leq \mu_{ph}, \quad \forall p \in \Pi, \forall h \in H_p \quad (3.2.7)$$

$$x : \text{binary}, \quad (w, \pi) \geq 0. \quad (3.2.8)$$

The objective function (3.2.1) is to maximize the net revenue considering multiple fare-classes for each path/itinerary. The cover constraints (3.2.2) and (3.2.3) distinguish between the set of mandatory and optional legs. Constraints (3.2.4) and (3.2.5) are respectively the conservation of flow and the aircraft resource count restrictions. Constraint (3.2.6) 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 (3.2.3) for any $j \in L^O$, then $\pi_{ph} = 0, \forall p \in \Pi(j)$. Constraint (3.2.7) 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 (3.2.8) imposes logical restrictions on the decision variables.

We assume that aircraft turn-times in (3.2.4) and availabilities in (3.2.5) are suitably adjusted to permit routine scheduled maintenance requirements, given any feasible solution to Model FSFAM1 (see Clarke et al. (1996), Talluri (1998), and 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 (3.2.4). This can be realized by linking any maintenance requirement with a particular flight leg. Furthermore, maintenance requiring 4-5 hours every 3-5 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 (3.2.5). However, due to the fact that 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, in order 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 Section 3.3) substantially enhances the solvability of the model by tightening its LP relaxation.

Toward this end, consider the following constraints:

$$z_p - \sum_{a \in AT} x_{aj} \leq 0, \quad \forall p \in \Pi^O, \forall j \in L^O(p) \quad (3.2.9)$$

$$z_p - \sum_{a \in AT} \sum_{j \in L^O(p)} x_{aj} \geq 1 - |L^O(p)|, \quad \forall p \in \Pi^O \quad (3.2.10)$$

$$\sum_{a \in AT} x_{aj} \leq \sum_{p \in \Pi^O(j)} z_p, \quad \forall j \in L^O \quad (3.2.11)$$

$$\pi_{ph} \leq \mu_{ph} z_p, \quad \forall p \in \Pi^O, \forall h \in H_p \quad (3.2.12)$$

$$\pi_{ph} \leq \mu_{ph}, \quad \forall p \in \Pi \setminus \Pi^O, \forall h \in H_p \quad (3.2.13)$$

$$(x, z) : \text{binary}, \quad (w, \pi) \geq 0. \quad (3.2.14)$$

Constraint (3.2.9) ensures that if any leg is excluded from the network, then all paths that contain this leg will also be excluded, and Constraint (3.2.10) 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 (3.2.11) 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 (3.2.9), 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 (3.2.11) can help reduce the number of corresponding surviving x_{aj} -variables in the resulting subproblem. Constraints (3.2.12) and (3.2.13) partition (3.2.7) 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, (3.2.14) replaces (3.2.8) to include the binary restrictions on the z -variables. We shall refer to the model defined by (3.2.1)-(3.2.6) and (3.2.9)-(3.2.14) as **FSFAM2**, and we proceed now to further analyze and improve this augmented model.

3.3 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 Section 3.5 strongly justify the utility of these valid inequalities by demonstrating that they help to significantly improve the performance of the proposed algorithm.

3.3.1 Tightening Inequalities within Model FSFAM2

This section addresses the lifting of Constraints (3.2.6) and (3.2.12) within Model FSFAM2 via Propositions 3.1 and 3.2 below, respectively.

Proposition 3.1. The following is a valid inequality that can be used to replace Constraint (3.2.6):

$$\sum_{p \in \Pi(j)} \pi_{ph} \leq \sum_{a \in AT} \widetilde{Cap}_{ahj} x_{aj}, \quad \forall j \in L, h \in H, \quad (3.3.1)$$

where

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

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 a case of $j \in L^O$), then (3.3.1) is valid by (3.2.6). 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, (3.2.6) implies that $\sum_{p \in \Pi(j)} \bar{\pi}_{ph} \leq Cap_{a^*h}$, and (3.2.12)-(3.2.13) imply that $\sum_{p \in \Pi(j)} \bar{\pi}_{ph} \leq \sum_{p \in \Pi(j)} \mu_{ph}$, which yields $\sum_{p \in \Pi(j)} \bar{\pi}_{ph} \leq \widetilde{Cap}_{a^*hj}$, or that (3.3.1) is again valid. Moreover, (3.3.1) implies (3.2.6), even in the continuous sense, and hence can be used to replace it. \square

Henceforth, we shall assume that (3.2.6) has been replaced by the tighter inequality (3.3.1), which can particularly be useful when capacity exceeds demand in the sense that $Cap_{ah} > \sum_{p \in \Pi(j)} \mu_{ph}$ for some $a \in AT, h \in H, j \in L$. On the other hand, when demand exceeds capacity in the sense that $\mu_{ph} > \max_{a \in AT} Cap_{ah}$, for some $p \in \Pi^O, h \in H_p$, then the following result offers a tightening of (3.2.12).

Proposition 3.2. The following valid inequalities can be used to replace Constraint (3.2.12):

$$\pi_{ph} \leq \tilde{\mu}_{ph} z_p, \quad \forall p \in \Pi^O, h \in H_p, \quad (3.3.2)$$

where

$$\tilde{\mu}_{ph} \equiv \min\{\mu_{ph}, \max_{a \in AT} Cap_{ah}\}, \quad \forall p \in \Pi^O, h \in H_p.$$

Proof: For any feasible solution $(\bar{x}, \bar{z}, \bar{w}, \bar{\pi})$ to Model FSFAM2, and for any $p \in \Pi^O, h \in H_p$, if $\bar{z}_p = 0$, then (3.3.2) is valid by (3.2.12), and if $\bar{z}_p = 1$, then (3.2.12) implies that $\bar{\pi}_{ph} \leq \mu_{ph}$, and (3.2.6) along with (3.2.2)-(3.2.3) implies that $\bar{\pi}_{ph} \leq \max_{a \in AT} Cap_{ah}$, so that (3.3.2) is again valid. \square

3.3.2 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 3.3. The following are valid inequalities for Model FSFAM2:

$$\pi_{ph} \leq \sum_{a \in AT} \min\{\mu_{ph}, Cap_{ah}\} x_{aj}, \quad \forall p \in \Pi(j), j \in L, h \in H_p. \quad (3.3.3)$$

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 (3.3.3) is valid from (3.2.6). Otherwise, from

(3.2.2)-(3.2.3) 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, (3.2.6) implies that $\bar{\pi}_{ph} \leq Cap_{a^*h}$ and (3.2.12)-(3.2.13) imply that $\bar{\pi}_{ph} \leq \mu_{ph}$, which yields $\bar{\pi}_{ph} \leq \min\{\mu_{ph}, Cap_{a^*h}\}$, or that (3.3.3) holds true. \square

The next class of valid inequalities lifts (3.2.10) for any $p \in \Pi^O$ based on an alternative optional path $q \in \Pi^O$ that shares at least two optional legs with it.

Proposition 3.4. The following are valid inequalities for Model FSFAM2:

$$z_p - \sum_{a \in AT} \sum_{j \in L^O(p) \setminus L^O(q)} x_{aj} \geq z_q - |L^O(p) \setminus L^O(q)|, \quad \forall p \in \Pi^O, q \in \Pi^O, |L^O(p) \cap L^O(q)| \geq 2. \quad (3.3.4)$$

Moreover, (3.3.4) is implied by (3.2.9), (3.2.10), and $0 \leq z_q \leq 1$ when $|L^O(p) \cap L^O(q)| \leq 1$.

Proof: When $z_q = 0$ or $\sum_{a \in AT} x_{aj} = 0$ for any $j \in L^O(p) \setminus L^O(q)$, then (3.3.4) is implied by (3.2.3) and $z_p \geq 0$. When $z_q = 1$ and $\sum_{a \in AT} x_{aj} = 1, \forall j \in L^O(p) \setminus L^O(q)$, then (3.3.4) asserts that $z_p = 1$, which is again valid. Hence, (3.3.4) is valid for all $p \in \Pi^O, q \in \Pi^O$. However, when $|L^O(p) \cap L^O(q)| \leq 1$, we have from (3.2.10) and (3.2.9) that,

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

where the first inequality simply rewrites (3.2.10), the second inequality uses (3.2.9) (written for $q \in \Pi^O$), and the final inequality follows from the nonnegativity of the second term in the preceding equation. Hence, (3.3.4) is implied by the continuous relaxation to FSFAM2 in this case. \square

Remark 3.1. It is not likely useful to incorporate all the inequalities (3.3.3) or (3.3.4) *a priori* within Model FSFAM2. Rather, we can solve the LP relaxation of Model FSFAM2

and then include members of (3.3.3) and (3.3.4) that are violated at the resulting solution, if any, within the model at the root node before proceeding further with the algorithmic process. \square

Next, prompted by (3.2.9) and (3.2.11), 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

$$\nu^{j^*} = \min_{p \in \Pi^O(j^*)} |S_p^{j^*}|. \quad (3.3.5)$$

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

Proposition 3.5. For any $j^* \in L^{O+}$, let $S^{j^*} \subseteq S_U^{j^*} \equiv \cup_{p \in \Pi^O(j^*)} S_p^{j^*}$ 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:

$$\sum_{j \in S^{j^*}} \sum_{a \in AT} x_{aj} \geq \nu^{j^*} \sum_{a \in AT} x_{aj^*}. \quad (3.3.6)$$

Proof: Note that if $\sum_{a \in AT} x_{aj^*} = 0$ then (3.3.6) is trivially valid. Else, from (3.2.3), we have that $\sum_{a \in AT} x_{aj^*} = 1$, and so from (3.2.11), there exists a $p^* \in \Pi^O(j^*)$ for which $z_{p^*} = 1$. This in turn implies from (3.2.3) and (3.2.9) that $\sum_{a \in AT} x_{aj} = 1, \forall j \in S_{p^*}^{j^*}$. Since $|S^{j^*} \cap S_{p^*}^{j^*}| \geq \nu^{j^*}$ by the given hypothesis, we therefore have (3.3.6) holding true. \square

In order to generate a strong version of (3.3.6) for any given $j^* \in L^{O+}$, where ν^{j^*} is then computed via (3.3.5), we would like to utilize a set S^{j^*} as per Proposition 3.5 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 $p \in \Pi^O(j^*)$. The following routine is designed with this motivation.

Routine R1 for Generating (3.3.6)

Initialization: Given $S_{\cup}^{j^*}$ as defined in Proposition 3.5, let η_j be the number of times that leg j appears in the different sets $S_p^{j^*}$ for $p \in \Pi^O(j^*)$. Initialize the counter $\nu_p = 0, \forall p \in \Pi^O(j^*)$, and let $S^{j^*} = \phi$.

Step 1: *Extract* (i.e., select and remove) $\hat{j} \in S_{\cup}^{j^*}$ having the highest value of η_j for $j \in S_{\cup}^{j^*}$, and include \hat{j} within S^{j^*} . Furthermore, for each $p \in \Pi^O(j^*)$ such that $\hat{j} \in S_p^{j^*}$, increment $\nu_p \leftarrow \nu_p + 1$, and if this incremented value $\nu_p = \nu^{j^*}$, then replace $\eta_j \leftarrow \eta_j - 1, \forall j \in S_p^{j^*} \cap S_{\cup}^{j^*}$.

Step 2: If $\nu_p \geq \nu^{j^*}, \forall p \in \Pi^O(j^*)$, then generate (3.3.6) using the current set S^{j^*} . Else, return to Step 1.

Example 3.1. Suppose that $j^* = 1$ 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 $S_4^{j^*} = \{6, 7, 15\}$. Hence, we have $\nu^{j^*} = 2$, and $S_{\cup}^{j^*} = \{5, 6, 7, 8, 9, 10, 12, 15\}$, with the corresponding vector $\eta \equiv [\eta_j, j \in S_{\cup}^{j^*}]$ given by $\eta = \{2, 2, 3, 1, 1, 1, 1, 1\}$. We initialize the vector $\nu \equiv [\nu_p, p \in \Pi^O(j^*)]$ as $\nu = [0, 0, 0, 0]$, and let $S^{j^*} = \phi$.

Routine R1 then proceeds through the following loops:

- (1) Extract $j = 7$ from $S_{\cup}^{j^*}$, and set $S^{j^*} = \{7\}$ and $\nu = [0, 1, 1, 1]$.
- (2) Extract $j = 5$ from $S_{\cup}^{j^*}$, and set $S^{j^*} = \{7, 5\}$ and $\nu = [1, 2, 1, 1]$. Also, since $\nu_2 = 2 = \nu^{j^*}$, we set $\eta_8 = 0$ and $\eta_9 = 0$.
- (3) Extract $j = 6$ from $S_{\cup}^{j^*}$, and set $S^{j^*} = \{7, 5, 6\}$ and $\nu = [2, 2, 1, 2]$. Furthermore, based on $\nu_4 = 2$, we set $\eta_{15} = 0$. (Note that $\nu_1 = 2$ as well, but $S_1^{j^*} \cap S_{\cup}^{j^*} = \phi$.)
- (4) Extract $j = 10$ from $S_{\cup}^{j^*}$, and set $S^{j^*} = \{7, 5, 6, 10\}$ and $\nu = [2, 2, 2, 2]$.

Furthermore, based on $\nu_3 = 2$, we set $\eta_{12} = 0$. Since $\nu_p \geq \nu^{j^*}, \forall p \in \Pi^O(j^*)$, we terminate this process and generate (3.3.6) as

$$\sum_{a \in AT} [x_{a7} + x_{a5} + x_{a6} + x_{a10}] \geq 2 \sum_{a \in AT} x_{a1}. \quad \square$$

Alternatively, we can solve a *separation problem* to generate a valid inequality (3.3.6) 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 (3.3.5), let

$$\theta \equiv \nu^{j^*} \sum_{a \in AT} \bar{x}_{aj^*}. \quad (3.3.7)$$

We would now like to select $S^{j^*} \subseteq S_{\cup}^{j^*}$ as per Proposition 3.5 that minimizes the left-hand side of (3.3.6) for the current LP solution. Hence, defining binary variables

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

we can formulate the following separation problem (SEP1) to generate (3.3.6), where the objective function determines the left-hand side of (3.3.6) at the given LP solution, and the constraints represent the restrictions on S^{j^*} as per Proposition 3.5.

$$\mathbf{SEP1} : \text{Minimize} \quad \sum_{j \in S_{\cup}^{j^*}} \left[\sum_{a \in AT} \bar{x}_{aj} \right] y_j \quad (3.3.8)$$

subject to:

$$\sum_{j \in S_p^{j^*}} y_j \geq \nu^{j^*}, \quad \forall p \in \Pi^O(j^*) \quad (3.3.9)$$

$$y : \text{binary}. \quad (3.3.10)$$

If the optimal objective value to Problem SEP1 (or any heuristic feasible solution value) is less than θ as given by (3.3.7), then the corresponding cut (3.3.6) 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 3.2. 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 (3.2.3) and (3.2.9)-(3.2.11) 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 (3.3.8)-(3.3.10) is then given as follows:

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

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 as for Example 3.1, which happens to delete the given LP solution. \square

The following section explores conditions under which (3.3.6) is facet-defining with respect to a particular substructure of Model FSFAM2.

3.3.3 Partial Convex Hull Representations and Related Facets and Separation Routines

Given any $j^* \in L^{O+}$, define ν^{j^*} as in (3.3.5). Consider the set of constraints defined by (3.2.3), (3.2.9), and (3.2.11). Noting the appearance of the x -variables in these constraints, let us define the aggregate variables

$$\xi_j \equiv \sum_{a \in AT} x_{aj}, \quad \forall j \in L^O. \quad (3.3.11)$$

Now, for the given j^* , consider the substructure defined by (3.2.3), (3.2.9), and (3.2.11)

under the change of variables (3.3.11) as given by

$$z_p - \xi_j \leq 0, \quad \forall p \in \Pi^O(j^*), \forall j \in L^O(p) \quad (3.3.12)$$

$$\xi_{j^*} \leq \sum_{p \in \Pi^O(j^*)} z_p \quad (3.3.13)$$

$$0 \leq \xi_j \leq 1, \quad \forall j \in L^O(p), p \in \Pi^O(j^*); \quad z_p \text{ binary}, \quad \forall p \in \Pi^O(j^*). \quad (3.3.14)$$

Note that by virtue of (3.2.3), (3.2.14), and (3.3.11), although the ξ_j -variables are binary-valued, we have declared them to be continuous on $[0, 1]$ in (3.3.12)-(3.3.14) since it is easily verified that these variables are automatically binary-valued at extreme point solutions to (3.3.12)-(3.3.14) 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 (3.3.12) since $j^* \in L^O(p), \forall p \in \Pi^O(j^*)$ by the definition of $\Pi^O(j^*)$. Moreover, if $\xi_{j^*} = 1$, then the system (3.3.12)-(3.3.14) 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 (3.3.12)-(3.3.14) as the following disjunction in terms of just the ξ_j -variables:

$$\bigvee_{p \in \Pi^O(j^*)} \{ \xi_j \geq \xi_{j^*}, \forall j \in S_p^{j^*}; \quad 0 \leq \xi_j \leq 1, \forall j \in S_p^{j^*} \cup \{j^*\} \}, \quad (3.3.15)$$

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

Proposition 3.6. The convex hull of (3.3.15) is given by

$$C(j^*) = \{ (\xi_j, \lambda_p) : \xi_j^p \geq \xi_{j^*}^p, \quad \forall j \in S_p^{j^*}, \forall p \in \Pi^O(j^*) \quad (3.3.16)$$

$$0 \leq \xi_j^p \leq \lambda_p, \quad \forall j \in S_p^{j^*} \cup \{j^*\}, \forall p \in \Pi^O(j^*) \quad (3.3.17)$$

$$\sum_{p \in \Pi^O(j^*)} \lambda_p = 1 \quad (3.3.18)$$

$$\xi_j = \sum_{p \in \Pi^O(j^*)} \xi_j^p, \quad \forall j \in S_{\cup}^{j^*} \cup \{j^*\}. \quad (3.3.19)$$

Proof: Follows directly from Balas (1974) (or see Sherali and Shetty (1980)). \square

Proposition 3.7. The facet-defining inequalities (or simply, facets) of $C(j^*)$ are of the type

$$\sum_{j \in S_{\cup}^{j^*}} \gamma_j \xi_j \geq \gamma_{j^*} \xi_{j^*} + \gamma_o, \quad (3.3.20)$$

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

$$\gamma_j \geq \alpha_j^p - \beta_j^p, \quad \forall j \in S_p^{j^*}, \quad p \in \Pi^O(j^*) \quad (3.3.21)$$

$$\gamma_{j^*} \leq \sum_{j \in S_p^{j^*}} \alpha_j^p + \beta_{j^*}^p, \quad \forall p \in \Pi^O(j^*) \quad (3.3.22)$$

$$\sum_{j \in S_p^{j^*} \cup \{j^*\}} \beta_j^p + \gamma_o = 0, \quad \forall p \in \Pi^O(j^*) \quad (3.3.23)$$

$$(\alpha, \beta) \geq 0. \quad (3.3.24)$$

Proof: Consider $C(j^*)$ and denote respective dual multipliers α_j^p associated with each constraint in (3.3.16); β_j^p associated with each constraint $\lambda_p - \xi_j^p \geq 0$ in (3.3.17); γ_o with (3.3.18); γ_j with each constraint in (3.3.19) written as $\xi_j - \sum_{p \in \Pi^O(j^*)} \xi_j^p = 0$ for $j \in S_{\cup}^{j^*}$, and γ_{j^*} with the constraint $-\xi_{j^*} + \sum_{p \in \Pi^O(j^*)} \xi_{j^*}^p = 0$ in (3.3.19) (for $j = j^*$).

Then, by duality (or Farkas' Lemma), the projection of $C(j^*)$ onto the space of the (original) variables $(\xi_j, j \in S_{\cup}^{j^*} \cup \{j^*\})$ is given by the set of constraints (3.3.20), where (γ, γ_o) along with (α, β) satisfy the dual feasibility constraints (3.3.21), (3.3.22), and (3.3.23), written respectively with respect to the columns of the variables $\{\xi_j^p, \forall j \in S_p^{j^*}, p \in \Pi^O(j^*)\}$, $\{\xi_{j^*}^p, \forall p \in \Pi^O(j^*)\}$, and $\{\lambda_p, \forall p \in \Pi^O(j^*)\}$, and where (3.3.24) records the nonnegativity restrictions on the (α, β) -variables. Moreover, note that the cone defined by (3.3.21)-(3.3.24) is pointed with the vertex at the origin since setting all the constraints in this set as equalities yields the origin as the unique solution. Hence, by Balas (1974) (or see Sherali and Shetty (1980)), the facets of $C(j^*)$ are given by (3.3.20) corresponding to extreme directions of the pointed cone defined by (3.3.21)-(3.3.24). \square

Remark 3.2. Propositions 3.6 and 3.7 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 3.6 (with the ξ_j -variables replaced by the x -variables using (3.3.11)) for some particular indices $j^* \in L^{O+}$, e.g., those for which the disjunction (3.3.15) 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 (3.3.11), and if this $\bar{\xi}$ violates any disjunction given by (3.3.15), we can solve the following separation problem to possibly delete \bar{x} , based on Proposition 3.7:

$$\mathbf{SEP2} : \text{Minimize} \quad \sum_{j \in S_U^{j^*}} \bar{\xi}_j \gamma_j - \bar{\xi}_{j^*} \gamma_{j^*} - \gamma_o \quad (3.3.25)$$

$$\text{subject to:} \quad (3.3.21) - (3.3.24) \quad (3.3.26)$$

$$\sum_{j \in S_U^{j^*}} \bar{\xi}_j \gamma_j - \bar{\xi}_{j^*} \gamma_{j^*} - \gamma_o \geq -1, \quad (3.3.27)$$

where (3.3.27) 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 (3.3.27)), we will have actually generated a facet of $C(j^*)$ (by Proposition 3.7), which deletes the current LP solution. \square

Finally, we address the issue raised in Remark 3.1 and identify certain sufficient conditions under which the valid inequality (3.3.6) given by Proposition 3.5 would be facet-defining for $C(j^*)$ as per Proposition 3.7.

Proposition 3.8. Consider the inequality (3.3.6) given by Proposition 3.5, which under (3.3.11), is restated as follows:

$$\sum_{j \in S^{j^*}} \xi_j \geq \nu^{j^*} \xi_{j^*}. \quad (3.3.28)$$

Suppose that

$$|S^{j^*} \cap S_p^{j^*}| = \nu^{j^*}, \forall p \in \Pi^O(j^*) \quad \text{and that} \quad |S^{j^*}| = |\Pi^O(j^*)|. \quad (3.3.29)$$

Moreover, suppose that the following equations are linearly independent:

$$\sum_{j \in S_p^{j^*} \cap S^{j^*}} \gamma_j = \nu^{j^*}, \quad \forall p \in \Pi^O(j^*). \quad (3.3.30)$$

Then (3.3.28) defines a facet of $C(j^*)$.

Proof: By Proposition 3.7, it is sufficient to show that under the conditions (3.3.29) and (3.3.30), we have that (3.3.28) is of the form (3.3.20) corresponding to some extreme direction of (3.3.21)-(3.3.24). To exhibit the latter, we identify next a set of active constraints in (3.3.21)-(3.3.24), which together with a normalization restriction, yields a unique feasible solution to (3.3.21)-(3.3.24) that produces the associated facet (3.3.20) as given by (3.3.28).

Toward this end, set $\beta \equiv 0$ (hence, $\gamma_o = 0$ from (3.3.23)) and set $\alpha_j^p = \gamma_j$, $\forall j \in S_p^{j^*} \cap S^{j^*}$ in (3.3.21) for all $p \in \Pi^O(j^*)$, and let $\alpha_j^p \equiv 0$ otherwise. Furthermore, designate all the constraints in (3.3.22) as active, which yields from the foregoing restrictions that

$$\gamma_{j^*} = \sum_{j \in S_p^{j^*} \cap S^{j^*}} \gamma_j, \quad \forall p \in \Pi^O(j^*). \quad (3.3.31)$$

Note that by the conditions stated in (3.3.29) and (3.3.30), for any fixed γ_{j^*} , the equations in (3.3.31) define $|S^{j^*}|$ linearly independent equations in $|S^{j^*}|$ unknowns (noting by the statement of Proposition 3.5 that $\cup_{p \in \Pi^O(j^*)} \{S_p^{j^*} \cap S^{j^*}\} = S^{j^*}$). Hence, for any γ_{j^*} , (3.3.31) yields a unique solution. Noting that $|S_p^{j^*} \cap S^{j^*}| = \nu^{j^*}$, $\forall p \in \Pi^O(j^*)$ by (3.3.29), this unique solution must be given by $\gamma_j = \gamma_{j^*} / \nu^{j^*}$, $\forall j \in S^{j^*}$ (since this yields a feasible solution to (3.3.31)). Using $\gamma_{j^*} = \nu^{j^*}$ as an arbitrary normalization restriction, we then get $\gamma_j = 1$, $\forall j \in S^{j^*}$ (with $\gamma_j = 0$, $\forall j \in S_U^{j^*} \setminus S^{j^*}$, by virtue of setting all the restrictions in (3.3.21) as active). Hence, along with the foregoing normalization constraint, the designated set of active constraints yield a unique feasible solution to (3.3.21)-(3.3.24) as above, where the corresponding facet-defining inequality (3.3.20) is given by (3.3.28). \square

Example 3.3. Consider the data of Example 3.1 and the inequality (3.3.6) generated via Routine R1, which in the form of (3.3.28), is given by:

$$\xi_5 + \xi_6 + \xi_7 + \xi_{10} \geq 2\xi_1, \quad (3.3.32)$$

where, $S^{j^*} \equiv \{5, 6, 7, 10\}$ and $\nu^{j^*} = 2$. Furthermore, (3.3.29) holds true, and (3.3.30) 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 3.8, (3.3.32) is facet-defining for $C(1)$. \square

3.4 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 Section 3.3, 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 then reiterate this process as needed.

Algorithm A1:

(Step A1.1) Tighten Model Representation: We begin by tightening Model FSFAM2 by replacing (3.2.6) and (3.2.12) with (3.3.1) and (3.3.2) using Propositions 3.1 and 3.2, respectively, and adding cuts of the type (3.3.6) via Routine R1. Next, we solve the LP relaxation of resulting model and use *rounds of cuts* given by Propositions 3.3, 3.4, 3.5, 3.6, and 3.7, using the separation problems SEP1 and SEP2 for the cases of Propositions 3.5 and

3.6-3.7, respectively, as well as possibly including the higher dimensional partial convex hull representations of Proposition 3.6.

Using $\xi \equiv (\xi_j, j \in L^O)$ as defined in (3.3.11), let us denote the cuts of the type (3.3.6) and (3.3.20) generated via SEP1 and SEP2, respectively, jointly as $A\xi \leq b$. Also, denote the generated valid inequalities of the type (3.3.3) via Proposition 3.3 as $Dx + E\pi \leq 0$.

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

$$\text{Model FSFAM}^+ : \quad \text{Maximize} \quad \sum_{p \in \Pi} \sum_{h \in H_p} f_{ph} \pi_{ph} - \sum_{a \in AT} \sum_{j \in L} c_{aj} x_{aj} \quad (3.4.1)$$

subject to :

$$\sum_{a \in AT} x_{aj} = 1, \quad \forall j \in L^M \quad (3.4.2)$$

$$\sum_{a \in AT} x_{aj} \leq 1, \quad \forall j \in L^O \quad (3.4.3)$$

$$\sum_{j \in L} b f_{jn} x_{aj} + \sum_{g \in G_a} b g_{gn} w_g = 0, \quad \forall n \in N_a, \forall a \in AT \quad (3.4.4)$$

$$\sum_{j \in CS_a} x_{aj} + \sum_{g \in CS_a} w_g \leq N A_a, \quad \forall a \in AT \quad (3.4.5)$$

$$z_p - \sum_{a \in AT} x_{aj} \leq 0, \quad \forall p \in \Pi^O, \forall j \in L^O(p) \quad (3.4.6)$$

$$z_p - \sum_{a \in AT} \sum_{j \in L^O(p)} x_{aj} \geq 1 - |L^O(p)|, \quad \forall p \in \Pi^O \quad (3.4.7)$$

$$\sum_{a \in AT} x_{aj} \leq \sum_{p \in \Pi^O(j)} z_p, \quad \forall j \in L^O \quad (3.4.8)$$

$$\sum_{p \in \Pi(j)} \pi_{ph} \leq \sum_{a \in AT} \widetilde{C} a_{pahj} x_{aj}, \quad \forall j \in L, \forall h \in H \quad (3.4.9)$$

$$\pi_{ph} \leq \tilde{\mu}_{ph} z_p, \quad \forall p \in \Pi^O, \forall h \in H_p \quad (3.4.10)$$

$$\pi_{ph} \leq \mu_{ph}, \quad \forall p \in \Pi \setminus \Pi^O, \forall h \in H_p \quad (3.4.11)$$

$$A\xi \leq b \quad (3.4.12)$$

$$Dx + E\pi \leq 0 \quad (3.4.13)$$

$$\xi_j = \sum_{a \in AT} x_{aj}, \quad \forall j \in L^O \quad (3.4.14)$$

$$(x, z) : \text{binary}, \quad (w, \pi) \geq 0. \quad (3.4.15)$$

(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 (as shown in Proposition 3.9, 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, (3.4.3) can then be eliminated in light of (3.4.14).

We next apply Benders decomposition to Model $\overline{\text{FSFAM}}^+$ by using the variables $(z, \xi) \equiv (z_p, \forall p \in \Pi^O, \text{ and } \xi_j, \forall j \in L^O)$ in the master program as binary restricted variables, and treating the remaining variables in the subproblem as continuous variables. By introducing a new variable, η_0 , to represent the value-function as computed via the subproblem given below for a fixed set of (z, ξ) -variables, the Benders master problem can be formulated as follows, where $BC(\theta, \Omega, \psi, \delta, \tau, \sigma)$ represents a Benders cut expression corresponding to the objective function of the dual subproblem derived below, and where Δ denotes the set of extreme points of the polyhedron defined by the constraints of this dual subproblem. Also, note that we have incorporated the aggregated plane-count constraint (3.4.21) within the master program, which is implied by $\sum_{j \in CS_a} x_{aj} + \sum_{g \in CS_a} w_g \leq NA_a, \forall a \in AT$ and $\xi_j = \sum_{a \in AT} x_{aj}, \forall j \in L^O$, and where $CS \equiv \cap_{a \in AT} CS_a$, to facilitate feasibility within the subproblem.

(Master Program)

$$\text{MP: Maximize } \eta_0$$

subject to:

$$\eta_0 \leq BC(\theta, \Omega, \psi, \delta, \tau, \sigma), \quad \forall (\theta, \Omega, \psi, \delta, \tau, \sigma) \in \Delta \quad (3.4.16)$$

$$z_p \leq \xi_j, \quad \forall p \in \Pi^O, \forall j \in L^O(p) \quad (3.4.17)$$

$$z_p - \sum_{j \in L^O(p)} \xi_j \geq 1 - |L^O(p)|, \quad \forall p \in \Pi^O \quad (3.4.18)$$

$$\xi_j \leq \sum_{p \in \Pi^O(j)} z_p, \quad \forall j \in L^O \quad (3.4.19)$$

$$A\xi \leq b, \quad (3.4.20)$$

$$\sum_{j \in CS \cap L^O} \xi_j \leq \sum_{a \in AT} NA_a, \quad (3.4.21)$$

$$z : \text{binary}, \quad 0 \leq \xi_j \leq 1, \forall j \in L^O, \quad \eta_0 : \text{unrestricted}. \quad (3.4.22)$$

Proposition 3.9. For any feasible solution $(\bar{z}, \bar{\xi})$ to Problem MP, we have that $\bar{\xi}$ is binary-valued.

Proof: For any $p \in \Pi^O$ and $j \in L^O(p)$, if $\bar{z}_p = 1$, then (3.4.17) and (3.4.22) imply that $\bar{\xi}_j = 1$. On the other hand, for any $j \in L^O$, if $\bar{z}_p = 0, \forall p \in \Pi^O(j)$, then (3.4.19) and (3.4.22) imply that $\bar{\xi}_j = 0$. Hence, $\bar{\xi}$ is binary-valued. \square

Given a solution $(\bar{z}, \bar{\xi})$ to Problem MP, i.e., given a feasible fleet assignment decision for all the optional legs and paths, we evaluate this solution via the following primal subproblem, where the dual variables are specified in parenthesis for each corresponding constraint for formulating the dual subproblem. Also, we have introduced an artificial vector $u \equiv (u_j, j \in L)$ with a large negative objective coefficient $-M$ associated with $e^T u$, where e is a conformable vector of ones, in order to ensure that the primal subproblem is feasible (the solution $(x, w, \pi) = 0$ then becomes trivially feasible). (This is done for computational expediency as justified by Mercier (2008); alternatively, we could include Benders feasibility cuts corresponding to the extreme directions of the polyhedron defined by the dual subproblem in addition to the Benders optimality cuts (3.4.16) - see Mercier (2008).) Thus, the accompanying dual subproblem is bounded in value and achieves an extreme point optimum for all $(\bar{z}, \bar{\xi})$ feasible to Problem MP.

(Primal Subproblem)

$$\mathbf{SP}(\bar{z}, \bar{\xi}): \quad \text{Maximize} \quad \sum_{p \in \Pi} \sum_{h \in H_p} f_{ph} \pi_{ph} - \sum_{a \in AT} \sum_{j \in L} c_{aj} x_{aj} - M e^T u$$

subject to:

$$\sum_{a \in AT} x_{aj} + u_j = 1, \quad \forall j \in L^M \quad (\theta_j) \quad (3.4.23)$$

$$\sum_{a \in AT} x_{aj} + u_j = \bar{\xi}_j, \quad \forall j \in L^O \quad (\theta_j) \quad (3.4.24)$$

$$\sum_{j \in L} b f_{jn} x_{aj} + \sum_{g \in G_a} b g_{gn} w_g = 0, \quad \forall n \in N_a, \forall a \in AT \quad (\Omega_{na}) \quad (3.4.25)$$

$$\sum_{j \in CS_a} x_{aj} + \sum_{g \in CS_a} w_g \leq N A_a, \quad \forall a \in AT \quad (\psi_a) \quad (3.4.26)$$

$$\sum_{p \in \Pi(j)} \pi_{ph} - \sum_{a \in AT} \widetilde{C a p}_{ahj} x_{aj} \leq 0, \quad \forall j \in L, \forall h \in H \quad (\delta_{jh}) \quad (3.4.27)$$

$$\pi_{ph} \leq \tilde{\mu}_{ph} \bar{z}_p, \quad \forall p \in \Pi^O, \forall h \in H_p \quad (\tau_{ph}) \quad (3.4.28)$$

$$\pi_{ph} \leq \mu_{ph}, \quad \forall p \in \Pi \setminus \Pi^O, \forall h \in H_p \quad (\tau_{ph}) \quad (3.4.29)$$

$$Dx + E\pi \leq 0, \quad (\sigma) \quad (3.4.30)$$

$$(x, w, \pi, v) \geq 0. \quad (3.4.31)$$

Writing $Dx + E\pi \leq 0$ in expanded form as

$$\sum_{a \in AT} \sum_{j \in L} D_{aj} x_{aj} + \sum_{p \in \Pi} \sum_{h \in H_p} E_{ph} \pi_{ph} \leq 0,$$

the corresponding dual to the above primal subproblem can be written as follows:

(Dual Subproblem)

$$\mathbf{DSP}(\bar{z}, \bar{\xi}): \quad \text{Minimize} \quad \sum_{j \in L^M} \theta_j + \sum_{j \in L^O} \bar{\xi}_j \theta_j + \sum_{a \in AT} N A_a \psi_a + \sum_{p \in \Pi^O} \sum_{h \in H_p} \tilde{\mu}_{ph} \bar{z}_p \tau_{ph} + \sum_{p \in \Pi \setminus \Pi^O} \sum_{h \in H_p} \mu_{ph} \tau_{ph}$$

subject to:

$$\theta_j + \sum_{n \in N_a} b f_{jn} \Omega_{na} + \psi_a - \sum_{h \in H} \widetilde{C} a p_{ahj} \delta_{jh} + D_{aj}^T \sigma \geq -c_{aj}, \quad \forall j \in L^M \cap CS_a, a \in AT \quad (3.4.32)$$

$$\theta_j + \sum_{n \in N_a} b f_{jn} \Omega_{na} - \sum_{h \in H} \widetilde{C} a p_{ahj} \delta_{jh} + D_{aj}^T \sigma \geq -c_{aj}, \quad \forall j \in L^M \setminus CS_a, a \in AT \quad (3.4.33)$$

$$\theta_j + \sum_{n \in N_a} b f_{jn} \Omega_{na} + \psi_a - \sum_{h \in H} \widetilde{C} a p_{ahj} \delta_{jh} + D_{aj}^T \sigma \geq -c_{aj}, \quad \forall j \in L^O \cap CS_a, a \in AT \quad (3.4.34)$$

$$\theta_j + \sum_{n \in N_a} b f_{jn} \Omega_{na} - \sum_{h \in H} \widetilde{C} a p_{ahj} \delta_{jh} + D_{aj}^T \sigma \geq -c_{aj}, \quad \forall j \in L^O \setminus CS_a, a \in AT \quad (3.4.35)$$

$$\sum_{n \in N_a} b g_{gn} \Omega_{na} + \psi_a \geq 0, \quad \forall g \in G_a \cap CS_a, a \in AT \quad (3.4.36)$$

$$\sum_{n \in N_a} b g_{gn} \Omega_{na} \geq 0, \quad \forall g \in G_a \setminus CS_a, a \in AT \quad (3.4.37)$$

$$\sum_{j \in L: p \in \Pi(j)} \delta_{jh} + \tau_{ph} + E_{ph}^T \sigma \geq f_{ph}, \quad \forall p \in \Pi, \forall h \in H_p \quad (3.4.38)$$

$$\theta_j \geq -M, \quad \forall j \in L \quad (3.4.39)$$

$$(\psi, \delta, \tau, \sigma) \geq 0, \quad (\theta, \Omega) : \text{unrestricted}. \quad (3.4.40)$$

Remark 3.3. Note that because of the inherent redundancy in the flow balance relationship (3.4.25), the dual polyhedron described by (3.4.32)-(3.4.40) contains a line and hence has no extreme points (see Bazaraa et al. (2005)). In order to resolve this technicality, we can set $\Omega_{na} \equiv 0$ for some single (arbitrarily selected) node n in each component subgraph for the flight network for each aircraft type $a \in AT$. Let us denote these zero-restrictions on the Ω -variables as $\Omega \in Z$, and let Δ denote the set of extreme points of the polyhedron defined by (3.4.32)-(3.4.40) along with $\Omega \in Z$. (This set Δ is nonempty because the resulting dual constraint set is feasible and, moreover, the homogenous system obtained by setting all the left-hand side constraint expressions equal to zero yields $(\theta, \Omega, \psi, \delta, \tau, \sigma) = 0$ as the unique solution.) Accordingly, the Benders' cut (3.4.16) for any $(\theta, \Omega, \psi, \delta, \tau, \sigma) \in \Delta$ is given by

$$\eta_0 \leq \sum_{p \in \Pi^O} \sum_{h \in H_p} [\tilde{\mu}_{ph} \tau_{ph}] z_p + \sum_{j \in L^O} \theta_j \xi_j + \left[\sum_{j \in L^M} \theta_j + \sum_{a \in AT} N A_a \psi_a + \sum_{p \in \Pi \setminus \Pi^O} \sum_{h \in H_p} \mu_{ph} \tau_{ph} \right].$$

Note that, in practice, we would solve the primal subproblem (3.4.23)-(3.4.31) using an LP

solver (e.g., CPLEX), and directly obtain a complementary dual optimal solution $(\theta, \Omega, \psi, \delta, \tau, \sigma)$ therefrom in order to derive the associated Benders cut. \square

(Step A1.3) Optimize a Restriction of Model FSFAM⁺: 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 FSFAM2, and label as mandatory all legs j in Π^O for which $\xi_j^* = 1$, while deleting 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 $\widehat{\Pi} \subseteq \Pi$ denote the set of paths resulting from fixing the z_p -variables, $\forall p \in \Pi^O$, and let $\widehat{L} \subseteq L$ 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 FSFAM2 (using Proposition 3.1, which remains valid):

$$\text{Maximize} \quad \sum_{p \in \widehat{\Pi}} \sum_{h \in H_p} f_{ph} \pi_{ph} - \sum_{a \in AT} \sum_{j \in \widehat{L}} c_{aj} x_{aj} \quad (3.4.41)$$

subject to:

$$\sum_{a \in AT} x_{aj} = 1, \quad \forall j \in \widehat{L} \quad (3.4.42)$$

$$\sum_{j \in \widehat{L}} b f_{jn} x_{aj} + \sum_{g \in G_a} b g_{gn} w_g = 0, \quad \forall n \in N_a, \forall a \in AT \quad (3.4.43)$$

$$\sum_{j \in CS_a \cap \widehat{L}} x_{aj} + \sum_{g \in CS_a} w_g \leq N A_a, \quad \forall a \in AT \quad (3.4.44)$$

$$\sum_{p \in \widehat{\Pi}(j)} \pi_{ph} \leq \sum_{a \in AT} \widetilde{C} a p_{ahj} x_{aj}, \quad \forall j \in \widehat{L}, \forall h \in H \quad (3.4.45)$$

$$\pi_{ph} \leq \mu_{ph}, \quad \forall p \in \widehat{\Pi}, \forall h \in H_p \quad (3.4.46)$$

$$x : \text{binary}, \quad (w, \pi) \geq 0. \quad (3.4.47)$$

Algorithm A2:

We adopt an identical approach to Algorithm A1, except that at Step A1.2, we solve the relaxed MIP given by Model $\overline{\text{FSFAM}^+}$ directly, in lieu of applying Benders decomposition to this problem.

3.5 Computational Experiments

The algorithms proposed in the previous section 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.

Remark 3.4. Due to the large size of *time-space networks*, for fleet assignment models, preprocessing steps have been used to reduce the network size before solving fleet assignment problems, thereby reducing the computational effort. As described in Hane et al. (1995), the idea is to take advantage of *hub-and-spoke* structure of the network where there is a high level of concentration of arrivals and departures between hubs and spokes at a given time period. For *node aggregation*, one or multiple consecutive arrival node(s) followed by one or multiple consecutive departure node(s) were consolidated into a single node representing a time line instead of a time point. This reduces the number of nodes as well as eliminates the ground arcs between the nodes of interest. For *node islands*, stations (e.g., spokes) with a sparse activity were considered, and if conservation of flow ensured that there were no aircraft on the ground arcs, then the corresponding ground arcs were removed. By removing these ground arcs, some of the nodes might become isolated (i.e., form *islands*), representing separate time lines where one or more aircraft can be on the ground. \square

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 4780 paths), D2 (428 flights, 2282 paths),

D3 (572 flights, 3646 paths), D4 (690 flights, 4888 paths), D5 (1016 flights and 10726 paths), D6 (1238 flights, 14103 paths), and D7 (1476 flights, 17121 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 3.5. Typically, there are several feasibility issues that arise in data sets, which were addressed as follows. First, the (selected) outgoing and incoming arcs need to be coordinated for each station in order to conserve flows. In context, due to 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 co-exists in the data set. This facilitates feasible solutions. However, in general, to accommodate a wider set of feasible solutions, we can create dead-head 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 dead-head flight can reach this station before the end of the day. Similar to the ground arcs, such dead-head 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 (3.2.5) (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 Section 3.2, we compared results from FSFAM1 and FSFAM2 using the data set D1 (with 188 mandatory legs and 126 optional legs) in order to assess the tightening effect of the additional constraints used in FSFAM2. Both models produced an optimal solution, but the CPU time required for Model FSFAM1 was about 8.4 hours whereas that for Model FSFAM2 was about 7.2 hours, thus reducing the solution time by about 15%. Next, we solved FSFAM1, FSFAM2, and FSFAM⁺ using the data set D4 (with 104 mandatory legs and 586 optional legs) with a 24 hour time limit. Both FSFAM2 and FSFAM⁺ were solved to optimality, with FSFAM2 consuming

22.8 hours while FSFAM⁺ took 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 while additionally tightening the LP relaxation value further by 0.84%.

3.5.1 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, and II-3(a,b). Case II-1 implements the valid inequalities (3.3.1), (3.3.2), (3.3.3), and (3.3.4) from Propositions 3.1, 3.2, 3.3, and 3.4, respectively; Case II-2 *additionally* implements the valid inequalities (3.3.6) from Proposition 3.5 that are generated via Routine R1 and the separation problem SEP1 (see (3.3.8)-(3.3.10)); and Case II-3 investigates the cuts of Propositions 3.6 and 3.7 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 (3.3.25)-(3.3.27) (Case II-3 (b)). More specifically, in generating cuts of type (3.3.3) and (3.3.4), we selected the 10 most violated cuts at each round of the LP relaxation solution. Similarly, cuts of type (3.3.6) and (3.3.20) 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 3.1: 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 hrs. | 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 hrs. | 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 hrs. | 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 hrs. | 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 hrs. | 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 hrs. | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| D7: | | | | | | | | |
| Rel LP | N/A | % gap | no int. | 5.22 | 4.30 | 5.83 | 5.15 | 0.00 |
| CPU(hr) | | Rel CPU | solution | 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 hrs. | 1.00 | 0.98 | 0.97 | 0.98 | 0.83 |

Table 3.1 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, the table 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, while 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 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.

Table 3.2: 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 |

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 3.2), as compared with the results in Table 3.1. 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, while 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.

3.5.2 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 3.3 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.

Table 3.3: Computational Results with Different Optimality Tolerances

| Problem | Opt. 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 |

As evident from Table 3.3, increasing the optimality tolerance gradually deteriorated the quality of the solution produced, while significantly shortening 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.

3.5.3 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 multi-dimensional knapsack problem:

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

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 $|L^{O*}|$:

$$\begin{aligned}
& \text{Maximize} && \sum_{p \in \Pi^O} \sum_{h \in H_p} f_{ph} \pi_{ph} - \sum_{a \in AT} \sum_{j \in L^O} c_{aj} x_{aj} \\
& \text{subject to :} && (3.2.3); (3.2.6); (3.2.9); (3.2.10); \\
& && (3.2.11) \text{ for } j \in L^O, h \in H; \text{ and } (3.2.12); \\
& && \sum_{j \in L^O} x_{aj} \leq NA_a, \quad \forall a \in AT \\
& && \sum_{a \in AT} \sum_{j \in L^O} x_{aj} = |L^{O*}| \\
& && (x, z) : \text{binary, } \pi \geq 0.
\end{aligned}$$

Table 3.4 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.

Table 3.4: Effect of Integration

| Problem | FSFAM ⁺ (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 |

The results in Table 3.4 indicate that the objective function values for both the sequentially solved models FAM* and FAM** worsened in comparison with the integrated model FSFAM⁺ (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, in order to find the most profitable $|L^{O*}|$ legs, about 4% of the total CPU time was used on average for the case of FAM*, while 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 FSFAM⁺ 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 FSFAM⁺ over FAM* and FAM**, respectively.

3.5.4 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 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/A2, we fix the x -variables that have fractional values exceeding 0.9 to one, but we keep free the x -variables that result in values of one. 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 3.5 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

Table 3.5: 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 |

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.

Chapter 4

An Integrated Schedule Design and Fleet Assignment Model

4.1 Introduction

In this chapter, we present and analyze a model that integrates flight scheduling and fleet assignment, while considering flexible flight times, schedule balance, and recapture issues in addition to optional legs, path/itinerary-based demands, and multiple fare-classes. The formulated model is enhanced by augmenting it with several classes of valid inequalities, and a suitable application of Benders decomposition is used to facilitate its solution. Extensive computational results are presented to study the effect of the different modeling features and to demonstrate the efficacy of the proposed solution approach. Because of the massive size of an industrial nation-wide problem, we envisage the application of our model to improve operations within a regional section of the domestic market, while for example, considering mainline operations involving large (wide-body) aircraft.

In this connection, we note that Lohatepanont and Barnhart (2004) have also studied the simultaneous consideration of schedule design and fleet assignment in their integrated model,

while examining the effect of leg selections on path demands by including different demand correction terms and revising them iteratively using a schedule evaluation model. Due to the computational difficulty associated with estimating demands based on accepted itineraries at every iteration, they modified their approach to instead consider demand recapture rates that are approximately re-adjusted on the fly to reflect demand variations. In addition, the main model feeds leg selections and fleet assignment decisions along with itinerary demands into a passenger mix model as discussed in Kniker (1998), which is solved subsequently. Instead, we consider the flow of passengers along itineraries over the network together with flight scheduling and fleet assignment decisions in order to maximize profits. After obtaining the final schedule via our model, we can likewise use an appropriate schedule evaluation package as in Lohatepanont and Barnhart (2004) to more accurately assess the effect of the derived solution on demand, and reiterate as necessary.

The remainder of this chapter is organized as follows. We begin in Section 4.2 by presenting our proposed integrated schedule planning and fleet assignment mixed-integer programming model. Section 4.3 explores various classes of valid inequalities to tighten the model representation. Section 4.4 addresses algorithmic approaches that generate suitable members of the proposed classes of valid inequalities through related separation routines, and then subsequently apply Benders decomposition to efficiently optimize the developed model. Results from our computational experiments are reported and analyzed in Section 4.5.

4.2 Notation and Model Formulation

Airline schedule planners expand or reduce airline networks depending on market conditions, current economic situation, as well as from one season to another. One way to effect an acceptable modification to an existing complex airline network (nation-wide or regional) is to consider mandatory flight legs that have to be included in the flight schedule, along with a suitable set of optional flight legs that may or may not be included. It follows that the

deletion of any optional leg within a path implies that the path along with its corresponding potential demand would be excluded from consideration, regardless of the status of any remaining legs on that path. Of course, legs could possibly belong to multiple paths in this context.

In a tentative schedule, if some flight departure times are not fixed and can vary within suitable time-windows, then we can include multiple discretized copies of the corresponding flight arcs in each fleet type's time-space network within such time-windows, similar to the idea used in Rexing et al. (2000). As indicated by Jiang and Barnhart (2009), the resultant added flexibility can provide further flight connection opportunities that can positively impact fleet assignment decisions, particularly in de-peaked hub-and-spoke flight networks. (We refer the reader to these two foregoing papers as well as to Hane et al. (1995) for an illustration of the underlying airline network including leg copies.) Accordingly, let L^M and L^O (indexed by k) denote the *parent sets* of mandatory and optional legs, respectively, and for each $k \in L^M \cup L^O$, let U_k be the set of legs $j \in L$ that represent copies of leg k , where $|U_k| \geq 1$, and where L denotes the set of all such possible mandatory and optional leg copies. As per guidelines from *United Airlines*, in practice, it is sufficient to consider a time-window ranging from five to ten minutes on either side of an expected flight time. Furthermore, as evident from the computational experience reported in Jiang and Barnhart (2009), a coarse discretization in which the timings of the leg copies are five minutes apart provides adequately competitive schedules in comparison with finer discretizations. Adopting such a strategy in which each $|U_k|$ is at most three or five (corresponding respectively to 10 or 20 minute time-windows), permits us to design a model using path-based variables related to itineraries composed of suitable compositions of leg copies. Although most itineraries involve only one to two leg copies, unlike the model of Jiang and Barnhart (2009), we are able to accommodate arbitrary multiple leg itineraries, and moreover, this modeling approach facilitates the consideration of recapture issues as discussed below. In addition, we consider optional legs and different fare-classes as well in our model. In this context, note that the accompanying itinerary or path compositions and their associated expected demands depend

on particular combinations of leg copies and their related timings. For example, we cannot compose a path having two flight legs where the arrival time of the copy of the first leg is too late to connect to some copy of the second leg on that path. Also, in order to incorporate more realistic demand in our model, we consider a decrease in passenger itinerary demands as the departure time of a flight leg copy is set further away from an ideal departure time within a given time-window. Hence, the model seeks to appropriately select flight timings to compose promising itinerary connections that maximize overall profits.

Furthermore, considering profit alone may result in an unbalanced daily schedule. For example, if flight legs between a pair of stations are all optional, the solution from the above type of model may schedule several flights in the morning and none in the evening. If a balanced schedule over the day is also desirable for the purpose of inducing an improved market share and consumer satisfaction, as well as from a workload viewpoint, then we can collect flights that fall within each specified time range into a group, and accordingly require that a minimal or maximal number of flights must be selected from each such group. In our model, R represents the set of flight groups thus composed, indexed by r , where group r must offer at least n_r^l , but no more than n_r^u , flights, $\forall r \in R$.

As alluded above, our model also incorporates recapture considerations whereby demand lost on some path is re-accommodated by the airline on another path between the same (or *compatible*, i.e., close enough) origin and destination. This is particularly relevant here because of the alternative itineraries afforded by the varying flight schedule choices. This recapture feature can be modeled using *substitution factors* that represent the proportions of excess (i.e., *spilled*) demand on each path p that *can possibly be* routed to each of several alternative paths, along with corresponding decision variables that represent the portion of demand attributed to each path that is flown on every other admissible path. Note that our choice of path-based flow variables facilitates this modeling construct. In this context, the recapture rates are computed using the *Quality of Service Index* (QSI), which is a mechanism to forecast passenger behavior by quantifying the relative quality (or attractiveness) of different flight services, and is used to assess a market share by considering factors such as

the departure time of day, aircraft type, number of stops, travel time, and flight frequency (see Kniker (1998) and Barnhart et al. (2002)). We also note here that, for the sake of simplicity, we ignore recapture across different fare-classes as well as between different airlines. Whereas these additional types of demand recapture can be likewise accommodated within our proposed model, their incorporation requires more detailed data pertaining to consumer preferences as well as other airlines' costs and operations.

Prior to formulating our model, we present a glossary of notation used in this chapter, which 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), and Sherali et al. (2006)).

Sets and Indices:

AT : set of aircraft types, indexed by a .

L^M : *parent set* of mandatory legs (possibly having flexible departure times), indexed by k .

L^O : *parent set* of original optional legs (possibly having flexible departure times), but subject to possible deletion, indexed by k .

L : set of flight legs in the flight schedule, indexed by j (note that this includes possibly multiple copies of legs in $L^M \cup L^O$ as identified next).

U_k : set of legs $j \in L$ that represent copies of leg $k \in L^M \cup L^O$ (note that $|U_k| \geq 1$, where exactly one of these copies must be selected for each $k \in L^M$, and at most one of these copies must be selected for each $k \in L^O$).

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

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

CS_a : set of flight legs (in L) and ground arcs (in G_a) passing forward in time through a counting time-line in the (time-space) network for aircraft type a , $a \in AT$. (Hence, the total flow on these arcs in CS_a would represent the total number of aircraft of type a that are in active use within the cyclic network as measured at the particular time-line.)

Π : set of all paths (related to considered itineraries that are feasible with respect to connection times), indexed by p or q .

$\Pi(j)$: set of paths in Π that pass through leg j , $\forall j \in L$.

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

R : set of flight groups, indexed by r .

L_r : set of legs $j \in L$ corresponding to flights in group r .

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

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

Π_{ph} : set of paths to which demand in fare-class $h \in H_p$ on path $p \in \Pi$ can be possibly re-routed, indexed by p or q (note that $p \in \Pi_{ph}$; furthermore, $q \in \Pi_{ph} \Rightarrow h \in H_q$ and that the origin and destination for path q is the same (or compatible with) that for path p).

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 , $\forall a \in AT, h \in H$.

n_r^l (n_r^u) : minimal (maximal) number of flights offered from flight group r , $\forall r \in R$.

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

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

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

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

β_{pqh} : maximum proportion of spilled demand in fare-class $h \in H_p \cap H_q$ on path $p \in \Pi$ that can be possibly re-routed to a path $q \in \Pi$ having the same (or compatible) origin and destination, $q \neq p$, given that both paths p and q are activated (note that $\Pi_{ph} = \{p\} \cup \{q \in \Pi, q \neq p : h \in H_q \text{ and } \beta_{pqh} > 0\}$, $\forall p \in \Pi, h \in H_p$).

δ : penalty or re-booking cost per passenger recaptured (or re-routed) on an alternative path. (Whereas the model can readily accommodate a corresponding parameter δ_{pqh} similar to β_{pqh} , we assume for the sake of simplicity that δ depends only on the re-booking system and is independent of paths or fare-classes; in practice, this is typically \$50 to \$100.)

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 the network for aircraft type a , $\forall g \in G_a, a \in AT$.

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

π_{pqh} : number of passengers in fare-class h on path p who are accepted on path q , $\forall h \in H_p \cap H_q, p \in \Pi, q \in \Pi_{ph}$ (given that both paths p and q are activated).

Model Formulation:

Our proposed integrated *schedule design and fleet assignment model* (**SDFAM**) formulated below combines the schedule planning and fleet assignment processes, while accommodating additional features such as path/itinerary-based demands, flexible flight times, schedule balance, recapture, and multiple fare-classes.

$$\text{SDFAM : Maximize } \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}} f_{qh} \pi_{pqh} - \sum_{a \in AT} \sum_{j \in L} c_{aj} x_{aj} - \delta \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \pi_{pqh} \quad (4.2.1)$$

subject to:

$$\sum_{a \in AT} \sum_{j \in U_k} x_{aj} = 1, \quad \forall k \in L^M \quad (4.2.2)$$

$$\sum_{a \in AT} \sum_{j \in U_k} x_{aj} \leq 1, \quad \forall k \in L^O \quad (4.2.3)$$

$$\sum_{j \in L} b f_{jn} x_{aj} + \sum_{g \in G_a} b g_{gn} w_g = 0, \quad \forall n \in N_a, \forall a \in AT \quad (4.2.4)$$

$$\sum_{j \in CS_a \cap L} x_{aj} + \sum_{g \in CS_a \cap G_a} w_g \leq N A_a, \quad \forall a \in AT \quad (4.2.5)$$

$$z_p - \sum_{a \in AT} x_{aj} \leq 0, \quad \forall p \in \Pi, \forall j \in L(p) \quad (4.2.6)$$

$$z_p - \sum_{a \in AT} \sum_{j \in L(p)} x_{aj} \geq 1 - |L(p)|, \quad \forall p \in \Pi \quad (4.2.7)$$

$$\sum_{a \in AT} x_{aj} \leq \sum_{p \in \Pi(j)} z_p, \quad \forall j \in L \quad (4.2.8)$$

$$n_r^l \leq \sum_{a \in AT} \sum_{j \in L_r} x_{aj} \leq n_r^u, \quad \forall r \in R \quad (4.2.9)$$

$$\sum_{p \in \Pi} \sum_{q \in \Pi_{ph} \cap \Pi(j)} \pi_{pqh} \leq \sum_{a \in AT} C a p_{ah} x_{aj}, \quad \forall j \in L, \forall h \in H \quad (4.2.10)$$

$$\sum_{q \in \Pi_{ph}} \pi_{pqh} \leq \mu_{ph} z_p, \quad \forall p \in \Pi, \forall h \in H_p \quad (4.2.11)$$

$$\pi_{pqh} \leq \beta_{pqh} (\mu_{ph} - \pi_{pph}) z_q, \quad \forall p \in \Pi, \forall h \in H_p, q \in \Pi_{ph}, q \neq p \quad (4.2.12)$$

$$(x, z) \text{ binary}, (w, \pi) \geq 0. \quad (4.2.13)$$

The objective function in (4.2.1) is to maximize the net profit as given by the revenue (first term) minus the cost (next two terms), while considering multiple fare-classes in each path/itinerary, and where the third penalty term automatically induces an optimal solution to route passengers on their first path choices whenever possible. Constraints (4.2.2)-(4.2.5)

represent standard coverage, flow balance, and resource count restrictions (see Lohatepanont and Barnhart (2004), for example). Specifically, Constraints (4.2.2) and (4.2.3) are the coverage restrictions considering flexible flight times for the mandatory and optional legs, respectively. These constraints require that for each parent mandatory leg identified by $k \in L^M$, exactly one copy from $j \in U_k$ must be selected by assigning an aircraft to it, whereas for each parent optional leg identified by $k \in L^O$, at most one leg copy $j \in U_k$ should be selected. Constraints (4.2.4) and (4.2.5) are respectively the conservation of flow and the aircraft resource count restrictions. Constraint (4.2.6) ensures that if any leg copy is excluded from the network, then all paths that contain this leg copy will also be excluded, and Constraint (4.2.7) ensures that if all the leg copies contained in a path are included within the flight network, then the path that contains them is also activated. Constraint (4.2.8) relates the leg copies to the corresponding paths, asserting that if any leg copy is selected, then at least one path that includes this leg copy must also be activated. Constraint (4.2.9) promotes a balanced schedule. Constraint (4.2.10) 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. Constraint (4.2.11) restricts the average number of passengers accepted from each activated path (including those recaptured on another path) to be no more than the expected demand on that path for each fare-class. Moreover, if any path is not activated, then all its potential demand is nullified. However, note that this demand might yet be served in effect on some other activated path composed of alternative copies of identical parent legs. The diversion of spilled demand (i.e., recapture) is explicitly considered in (4.2.12), where, together with (4.2.11), we have that $\pi_{pqh} > 0$ only if both the paths p and q are activated, i.e., $z_p = z_q = 1$. Observe that Constraint (4.2.12) is nonlinear, but will be subsequently linearized in our model reformulation process (see Proposition 4.1 in Section 4.3). Finally, Constraint (4.2.13) imposes logical restrictions on the decision variables.

Note that in this model, we implicitly include maintenance considerations by assuming that aircraft turn-times in (4.2.4) and availabilities in (4.2.5) are suitably adjusted to permit routine scheduled maintenance requirements. For instance, by appropriately decreasing the

number of available aircraft of each type with respect to Constraint (4.2.5), extensive FAA-mandated scheduled maintenance checks such as Type C and Type D that require aircraft to be taken out of service for a long period of time can be (approximately) accommodated. If maintenance needs to be done for four to five hours every three to five days (Type A), it is assumed to be conducted overnight, as is typically done in practice for inactive short-haul aircraft. This would involve requiring a minimum number of aircraft of each type to overnight at a maintenance base. For maintenance that requires a short period of time, for example on a flight basis, we assume that the required duration is built into the turn-times within Constraint (4.2.4). In the same spirit, as in Barnhart et al. (1998), a maintenance arc can be generated connecting an end-of-day flight arrival time at a maintenance base to a ground arc node for each aircraft type, with associated time equal to the maintenance time, to route all resident aircraft through an overnight maintenance process. Furthermore, we can create a suitable number of mandatory *maintenance legs* and group them into a set for each aircraft type in a time-space network, given the minimum number of aircraft of each type that need a particular type of maintenance. However, since the fleet assignment model assigns aircraft types, instead of individual aircraft, to flight legs, it can only provide a desired number of maintenance opportunities without ensuring that the interval between maintenance visits is appropriately spaced for each aircraft. Therefore, given the scope of the model, the foregoing strategies only approximately accommodate maintenance requirements. In order 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 (see Cordeau et al. (2001), Desaulniers et al. (1997), Mercier et al. (2005), and Papadakos (2009)).

Also, note that we have assumed the same-every-day schedule and fleet assignment, and the cyclic time-space networks are accordingly built to repeat daily as described in Hane et al. (1995) and Sherali et al. (2006). If different day-of-week (DOW) demand patterns exist, the network can be built to cycle every multiple days, that is, the different daily time-space networks can be concatenated before linking the last events with the first events using wrap-around arcs. The solution will then yield different DOW schedules and fleet assignments.

Naturally, the size and complexity of the model would greatly increase in such a case and would require tailored solution approaches.

4.3 Valid Inequalities

In this section, we utilize the Reformulation-Linearization Technique (RLT) (Sherali and Adams, 1990, 1994) in concert with polyhedral analyses to derive classes of valid inequalities for enhancing the solvability of Model SDFAM. To begin with, observe that we have a non-linear term, $\pi_{pph}z_q$, in Constraint (4.2.12) that needs to be linearized. By substituting π_{pphq} for $\pi_{pph}z_q$, the following result provides the required equivalent linearized representation.

Proposition 4.1. The following valid linear inequalities can be used to equivalently replace Constraint (4.2.12):

$$\pi_{pqh} \leq \beta_{pqh}\mu_{ph}z_q - \beta_{pqh}\pi_{pphq}, \quad \forall p \in \Pi, \forall h \in H_p, \quad q \in \Pi_{ph}, \quad q \neq p, \quad (4.3.1)$$

where we additionally have

$$\pi_{pphq} \geq \pi_{pph} - \mu_{ph}(1 - z_q) \text{ and } \pi_{pphq} \geq 0, \quad \forall p \in \Pi, \forall h \in H_p, \quad q \in \Pi_{ph}, \quad q \neq p. \quad (4.3.2)$$

Proof: By Sherali and Adams (1994), for each $p \in \Pi$ and $h \in H_p$, an equivalent linearization of (4.2.12) can be derived by substituting $\pi_{pphq} \equiv \pi_{pph}z_q$ in (4.2.12) for each $q \in \Pi_{ph}$, $q \neq p$, and including the following four bound-factor RLT products constructed from the bounding relationships $0 \leq \pi_{pph} \leq \mu_{ph}$ and $0 \leq z_q \leq 1$:

$$[(\mu_{ph} - \pi_{pph})(1 - z_q)]_l \geq 0; \quad [\pi_{pph}(1 - z_q)]_l \geq 0; \quad [(\mu_{ph} - \pi_{pph})z_q]_l \geq 0; \text{ and } [\pi_{pph}z_q]_l \geq 0, \quad (4.3.3)$$

where $[\cdot]_l$ denotes the linearization of $[\cdot]$ under the substitution $\pi_{pphq} \equiv \pi_{pph}z_q$. Of the four bound-factor RLT constraints in (4.3.3), the first and fourth yield Constraint (4.3.2). Note that when $z_q = 1$, then Constraint (4.3.1) implies that $\pi_{pqh} \leq \beta_{pqh}\mu_{ph} - \beta_{pqh}\pi_{pphq}$, and

Constraint (4.3.2) implies that $\pi_{pphq} \geq \pi_{pph}$, which jointly yield $\pi_{pqh} \leq \beta_{pqh}\mu_{ph} - \beta_{pqh}\pi_{pphq} \leq \beta_{pqh}(\mu_{ph} - \pi_{pph})$ as required by (4.2.12). On the other hand, when $z_q = 0$, then Constraints (4.2.13), (4.3.1), and (4.3.2), respectively assert that $\pi_{pqh} \geq 0$, $\pi_{pqh} \leq -\beta_{pqh}\pi_{pphq}$, and $\pi_{pphq} \geq 0$, which yield $\pi_{pphq} = \pi_{pqh} = 0$, as required by (4.2.12). Therefore, the second and third inequalities in (4.3.3) are not necessary in the present context, and (4.2.12) can be equivalently replaced by (4.3.1) and (4.3.2) in Model SDFAM. \square

Next, while focusing on certain constraint structures within Model SDFAM, we derive additional valid inequalities in order to tighten the model representation. To begin with, note that (4.2.10) and (4.2.13) imply that

$$\sum_{p \in \Pi(j)} \pi_{pph} \leq \sum_{a \in AT} Cap_{ah} x_{aj}, \quad \forall j \in L, h \in H. \quad (4.3.4)$$

Also, (4.2.11) and (4.2.13) imply that

$$\pi_{pph} \leq \mu_{ph} z_p, \quad \forall p \in \Pi, h \in H_p. \quad (4.3.5)$$

Propositions 4.2-4.4 below are a direct consequence of (4.3.4) and (4.3.5), and derive identical valid inequalities to those proposed in Chapter 3.

Proposition 4.2. The following is a valid inequality:

$$\sum_{p \in \Pi(j)} \pi_{pph} \leq \sum_{a \in AT} \widetilde{Cap}_{ahj} x_{aj}, \quad \forall j \in L, h \in H, \quad (4.3.6)$$

where

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

Proposition 4.3. The following is a valid inequality:

$$\pi_{pph} \leq \tilde{\mu}_{ph} z_p, \quad \forall p \in \Pi, h \in H_p, \quad (4.3.7)$$

where

$$\tilde{\mu}_{ph} \equiv \min\{\mu_{ph}, \max_{a \in AT} Cap_{ah}\}, \quad \forall p \in \Pi, h \in H_p. \quad \square$$

Note that, in light of (4.3.5), the inequality (4.3.7) is useful to incorporate within SDFAM only when $\mu_{ph} > \max_{a \in AT} Cap_{ah}$, for some $p \in \Pi, h \in H_p$.

Proposition 4.4. The following are valid inequalities for Model SDFAM:

$$\pi_{pph} \leq \sum_{a \in AT} \min\{\mu_{ph}, Cap_{ah}\} x_{aj}, \quad \forall p \in \Pi(j), j \in L, h \in H_p. \quad \square \quad (4.3.8)$$

Finally, we propose another class of valid inequalities based on the development in Chapter 3, which can be generated via separation routines as discussed below to further tighten the model representation. Consider

$$U^{O+} \equiv \{j \in \cup_{k \in L^O} U_k : |U^O(p)| \geq 2, \forall p \in \Pi(j)\}, \quad (4.3.9)$$

where $U^O(p)$ is the set of optional leg copies contained in path p . For any $j^* \in U^{O+}$, define

$$S_p^{j^*} \equiv U^O(p) \setminus \{j^*\}, \quad \forall p \in \Pi(j^*), \quad (4.3.10)$$

and compute

$$\nu^* = \min_{p \in \Pi(j^*)} |S_p^{j^*}|. \quad (4.3.11)$$

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

Proposition 4.5. Let $S^* \subseteq S^{j^*} \equiv \cup_{p \in \Pi(j^*)} S_p^{j^*}$ for any $j^* \in U^{O+}$, where $|S^* \cap S_p^{j^*}| \geq \nu^*, \forall p \in \Pi(j^*)$. Then the following is a valid inequality:

$$\sum_{j \in S^*} \sum_{a \in AT} x_{aj} \geq \nu^* \sum_{a \in AT} x_{aj^*}. \quad \square \quad (4.3.12)$$

We utilize a set S^* as per Proposition 4.5 for which $|S^*|$ is as small as possible, in order to generate a strong version of (4.3.12) for any given $j^* \in U^{O+}$, where ν^* is accordingly

computed via (4.3.11). This is achieved by preferentially selecting legs to include within S^* that repeatedly appear within the sets $S_p^{j^*}$ for $p \in \Pi(j^*)$. With this motivation, the following separation routine described in Chapter 3 is utilized to generate a valid inequality (4.3.12) that deletes a computed optimal (or any feasible) solution $(\bar{x}, \bar{z}, \bar{w}, \bar{\pi})$ to the LP relaxation of Model SDFAM. For this purpose, for a selected $j^* \in U^{O+}$ and its corresponding value ν^* given by (4.3.11), let

$$\theta \equiv \nu^* \sum_{a \in AT} \bar{x}_{aj^*}. \quad (4.3.13)$$

We now need to select $S^* \subseteq S^{j^*}$ as per Proposition 4.5 that minimizes the left-hand side of (4.3.12) for the current LP solution. Hence, defining binary variables

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

we can formulate the following *separation problem* (SEP) to generate (4.3.12), where the objective function determines the left-hand side of (4.3.12) at the given LP solution, and the constraints represent the restrictions on S^* as per Proposition 4.5:

$$\mathbf{SEP} : \text{Minimize} \quad \sum_{j \in S^{j^*}} \left[\sum_{a \in AT} \bar{x}_{aj^*} \right] y_j \quad (4.3.14a)$$

$$\text{subject to:} \quad \sum_{j \in S_p^{j^*}} y_j \geq \nu^*, \quad \forall p \in \Pi(j^*) \quad (4.3.14b)$$

$$y \text{ binary.} \quad (4.3.14c)$$

If the optimal objective value for Problem SEP is less than θ as given by (4.3.13), then the cut (4.3.12) generated via the corresponding y -solution will delete the specified LP solution. In our implementation, we generated cuts of type (4.3.12) by selecting the indices j^* that yielded the 10 most violated inequalities with respect to the LP relaxation, out of up to 30 potential candidates identified.

We mention here that the derived valid inequalities are peculiar to the particular structure of SDFAM. The valid inequalities (4.3.1) and (4.3.2) of Proposition 4.1 are based on Constraints (4.2.11)-(4.2.13); the valid inequalities (4.3.6)-(4.3.8) of Propositions 4.2-4.4 are based on Constraints (4.2.10), (4.2.11), and (4.2.13); and the valid inequality (4.3.12) of Proposition 4.5 is based on Constraints (4.2.3), (4.2.6)-(4.2.8), and (4.2.13). Naturally, these are transferable to other network design problems that share the indicated particular constraint structures.

4.4 Solution Approach

We adopt two algorithmic approaches to solve Model SDFAM that utilize the proposed valid inequalities with and without Benders decomposition, denoted respectively as Algorithms A1 and A2 below.

Algorithm A1 starts with the linearized version of SDFAM, then incorporates valid inequalities to tighten its representation, and next applies Benders decomposition to a suitable relaxation of the resulting model to fix leg selection and flight timing decisions, and finally recovers fleet assignment and demand allocation decisions. This process enables an integrated look-ahead viewpoint in making the flight scheduling decisions, while retaining computational tractability.

Algorithm A1:

(Step A1.1) Linearization and Tightening of Model Representation: We first linearize Model SDFAM by replacing Constraint (4.2.12) with the inequalities (4.3.1) and (4.3.2) from Proposition 4.1. Then the valid inequalities (4.3.6), (4.3.7), and (4.3.8) from Propositions 4.2, 4.3, and 4.4, respectively, are added to Model SDFAM. Next, we solve the LP relaxation of the resulting model and sequentially generate *rounds of cuts* (4.3.12) given by Proposition 4.5 via the separation problem (4.3.14). Define $\xi_j \equiv \sum_{a \in AT} x_{aj}$, $\forall j \in L$, and

denote the cuts generated via the separation routine (4.3.14) as $A\xi \leq b$, where $\xi \equiv (\xi_j, j \in L)$. In addition, whenever the flight network possesses a hub-and-spoke structure, as is typically the case in practice for most legacy carriers, we also impose a constraint that if an optional flight k is selected, then the corresponding optional return flight \vec{k} is selected as well (and vice versa, given that such a pair exists), i.e., $\sum_{j \in U_k} \xi_j = \sum_{j \in U_{\vec{k}}} \xi_j, \forall$ such paired flights $k, \vec{k} \in L^O$. Denote this set of constraints as $D\xi = d$. The enhanced Model SDFAM can then be rewritten as follows:

$$\text{SDFAM}^+ : \text{Maximize } \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}} f_{qh} \pi_{pqh} - \sum_{a \in AT} \sum_{j \in L} c_{aj} x_{aj} - \delta \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \pi_{pqh} \quad (4.4.1)$$

subject to:

$$\sum_{j \in U_k} \xi_j = 1, \quad \forall k \in L^M \quad (4.4.2)$$

$$\sum_{j \in U_k} \xi_j \leq 1, \quad \forall k \in L^O \quad (4.4.3)$$

$$\sum_{j \in L} b f_{jn} x_{aj} + \sum_{g \in G_a} b g_{gn} w_g = 0, \quad \forall n \in N_a, a \in AT \quad (4.4.4)$$

$$\sum_{j \in CS_a \cap L} x_{aj} + \sum_{g \in CS_a \cap G_a} w_g \leq N A_a, \quad \forall a \in AT \quad (4.4.5)$$

$$z_p \leq \xi_j, \quad \forall p \in \Pi, j \in L(p) \quad (4.4.6)$$

$$z_p - \sum_{j \in L(p)} \xi_j \geq 1 - |L(p)|, \quad \forall p \in \Pi \quad (4.4.7)$$

$$\xi_j \leq \sum_{p \in \Pi(j)} z_p, \quad \forall j \in L \quad (4.4.8)$$

$$n_r^l \leq \sum_{j \in L_r} \xi_j \leq n_r^u, \quad \forall r \in R \quad (4.4.9)$$

$$\sum_{p \in \Pi} \sum_{q \in \Pi_{ph} \cap \Pi(j)} \pi_{pqh} \leq \sum_{a \in AT} C a p_{ah} x_{aj}, \quad \forall j \in L, h \in H \quad (4.4.10)$$

$$\sum_{p \in \Pi(j)} \pi_{pph} \leq \sum_{a \in AT} \widetilde{C} a p_{ahj} x_{aj}, \quad \forall j \in L, h \in H \quad (4.4.11)$$

$$\sum_{q \in \Pi_{ph}} \pi_{pqh} \leq \mu_{ph} z_p, \quad \forall p \in \Pi, h \in H_p \quad (4.4.12)$$

$$\pi_{pph} \leq \tilde{\mu}_{ph} z_p, \quad \forall p \in \Pi, h \in H_p : \mu_{ph} > \max_{a \in AT} Cap_{ah} \quad (4.4.13)$$

$$\pi_{pph} \leq \sum_{a \in AT} \hat{\mu}_{pha} x_{aj}, \quad \forall p \in \Pi(j), j \in L, h \in H_p, \quad (4.4.14)$$

$$\text{where } \hat{\mu}_{pha} \equiv \min\{\mu_{ph}, Cap_{ah}\}$$

$$\pi_{pqh} + \beta_{pqh} \pi_{pphq} \leq \beta_{pqh} \mu_{ph} z_q, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph}, q \neq p \quad (4.4.15)$$

$$\pi_{pph} - \pi_{pphq} \leq \mu_{ph}(1 - z_q), \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph}, q \neq p \quad (4.4.16)$$

$$A\xi \leq b, \quad (4.4.17)$$

$$D\xi = d, \quad (4.4.18)$$

$$\xi_j = \sum_{a \in AT} x_{aj}, \quad \forall j \in L \quad (4.4.19)$$

$$(x, z) \text{ binary}, \quad 0 \leq \xi_j \leq 1, \quad (w, \pi) \geq 0. \quad (4.4.20)$$

(Step A1.2) Applying Benders Decomposition: Denote Model $\overline{\text{SDFAM}}^+$ as Model SDFAM^+ where the binary restrictions on the x -variables are relaxed. Note that due to (4.4.6), (4.4.8), and (4.4.20), as proven in Chapter 3, the binary restrictions on the z -variables imply that ξ will also be binary-valued for any feasible solution to $\overline{\text{SDFAM}}^+$. Hence, using the procedure described next, the derived solution to Model $\overline{\text{SDFAM}}^+$ has (z, ξ) integral.

To solve $\overline{\text{SDFAM}}^+$, we apply Benders decomposition by including the constraints that involve the variables (z, ξ) in the master program, and by letting the remaining constraints define the subproblem, with the Benders cuts thereby representing the value function implied by the subproblem (see Benders, 1962). This yields the following decomposition:

(Master Program)

$$\text{MP:} \quad \text{Maximize} \quad \eta_0$$

subject to:

$$\eta_0 \leq BC(\lambda, \alpha, \beta, \gamma, \tau, \sigma, \theta), \quad \forall (\lambda, \alpha, \beta, \gamma, \tau, \sigma, \theta) \in \Delta \quad (4.4.21)$$

$$\sum_{j \in U_k} \xi_j = 1, \quad \forall k \in L^M \quad (4.4.22)$$

$$\sum_{j \in U_k} \xi_j \leq 1, \quad \forall k \in L^O \quad (4.4.23)$$

$$\sum_{j \in CS \cap L} \xi_j \leq \sum_{a \in AT} NA_a, \quad (4.4.24)$$

$$z_p \leq \xi_j, \quad \forall p \in \Pi, \forall j \in L(p) \quad (4.4.25)$$

$$z_p - \sum_{j \in L(p)} \xi_j \geq 1 - |L(p)|, \quad \forall p \in \Pi \quad (4.4.26)$$

$$\xi_j \leq \sum_{p \in \Pi(j)} z_p, \quad \forall j \in L \quad (4.4.27)$$

$$n_r^l \leq \sum_{j \in L_r} \xi_j \leq n_r^u, \quad \forall r \in R \quad (4.4.28)$$

$$A\xi \leq b, \quad (4.4.29)$$

$$D\xi = d, \quad (4.4.30)$$

$$z : \text{binary}, \quad 0 \leq \xi_j \leq 1, \quad \forall j \in L, \quad \eta_0 : \text{unrestricted}. \quad (4.4.31)$$

Here, η_0 is the value function, which is given by the Benders cuts (4.4.21) based on the dual to the subproblem defined below, where Δ denotes the set of extreme points of the polyhedron defined by the constraints of this dual subproblem. In order to facilitate feasibility, as in Chapter 3, an aggregated plane-count constraint (4.4.24) has been incorporated within the master program, where $CS \equiv \cup_{a \in AT} CS_a$.

The primal subproblem is stated below, where an artificial variable vector $v \equiv (v_k, k \in L^M \cup L^O)$ has been introduced within the set of constraints (4.4.33) and (4.4.34) in order to assure feasibility, along with an associated sufficiently large negative coefficient $-M$ in the objective function. Note that this construct is adopted for the sake of convenience so that the standard off-the-shelf software (CPLEX) used to solve the subproblem always returns an optimal solution, and therefore only optimality cuts are generated. In cases where some artificial variables in the subproblem are positive at optimality, the corresponding Benders cut behaves essentially as a feasibility cut. (Sherali and Lunday (2010) discuss how the

actual associated feasibility cut can be extracted therefrom, but as documented in their reported results, this does not always offer a computational advantage.) We refer the reader to Remark 4.1 below for additional related comments.

(Primal Subproblem)

$$\text{PSP}(\bar{z}, \bar{\xi}) : \text{Maximize } \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}} f_{qh} \pi_{pqh} - \sum_{a \in AT} \sum_{j \in L} c_{aj} x_{aj} - \delta \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \pi_{pqh} - M e^T v \quad (4.4.32)$$

subject to:

$$\sum_{a \in AT} \sum_{j \in U_k} x_{aj} + v_k = 1, \quad \forall k \in L^M \quad (\lambda_k) \quad (4.4.33)$$

$$\sum_{a \in AT} \sum_{j \in U_k} x_{aj} + v_k = \sum_{j \in U_k} \bar{\xi}_j, \quad \forall k \in L^O \quad (\lambda_k) \quad (4.4.34)$$

$$\sum_{j \in L} b f_{jn} x_{aj} + \sum_{g \in G_a} b g_{gn} w_g = 0, \quad \forall n \in N_a, \forall a \in AT \quad (\alpha_{na}) \quad (4.4.35)$$

$$\sum_{j \in CS_a \cap L} x_{aj} + \sum_{g \in CS_a \cap G_a} w_g \leq N A_a, \quad \forall a \in AT \quad (\beta_a) \quad (4.4.36)$$

$$\sum_{p \in \Pi} \sum_{q \in \Pi_{ph} \cap \Pi(j)} \pi_{pqh} - \sum_{a \in AT} C a p_{ah} x_{aj} \leq 0, \quad \forall j \in L, \forall h \in H \quad (\gamma_{jh}^1) \quad (4.4.37)$$

$$\sum_{p \in \Pi(j)} \pi_{pph} - \sum_{a \in AT} \widetilde{C} a p_{ahj} x_{aj} \leq 0, \quad \forall j \in L, \forall h \in H \quad (\gamma_{jh}^2) \quad (4.4.38)$$

$$\sum_{q \in \Pi_{ph}} \pi_{pqh} \leq \mu_{ph} \bar{z}_p, \quad \forall p \in \Pi, \forall h \in H_p \quad (\tau_{ph}^1) \quad (4.4.39)$$

$$\pi_{pph} \leq \tilde{\mu}_{ph} \bar{z}_p, \quad \forall p \in \Pi, \forall h \in H_p : \mu_{ph} > \max_{a \in AT} C a p_{ah} \quad (\tau_{ph}^2) \quad (4.4.40)$$

$$\pi_{pph} - \sum_{a \in AT} \hat{\mu}_{pha} x_{aj} \leq 0, \quad \forall p \in \Pi(j), j \in L, h \in H_p \quad (\sigma_{pjh}) \quad (4.4.41)$$

$$\pi_{pqh} + \beta_{pqh} \pi_{pph} \leq \beta_{pqh} \mu_{ph} \bar{z}_q, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph}, q \neq p \quad (\theta_{pqh}^1) \quad (4.4.42)$$

$$\pi_{pph} - \pi_{pphq} \leq \mu_{ph} (1 - \bar{z}_q), \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph}, q \neq p \quad (\theta_{pqh}^2) \quad (4.4.43)$$

$$(x, w, \pi) \geq 0. \quad (4.4.44)$$

The corresponding dual to this primal subproblem is given as follows:

(Dual Subproblem)

$$\begin{aligned}
\text{DSP}(\bar{z}, \bar{\xi}) : \text{Minimize } & \sum_{k \in L^M} \lambda_k + \sum_{k \in L^O} \sum_{j \in U_k} \bar{\xi}_j \lambda_k + \sum_{a \in AT} NA_a \beta_a + \sum_{p \in \Pi} \sum_{h \in H_p} \mu_{ph} \bar{z}_p \tau_{ph}^1 \\
& + \sum_{\substack{p \in \Pi \\ : \mu_{ph} > \max_{a \in AT} Cap_{ah}}} \sum_{h \in H_p} \tilde{\mu}_{ph} \bar{z}_p \tau_{ph}^2 + \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \beta_{pqh} \mu_{ph} \bar{z}_q \theta_{pqh}^1 \\
& + \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \mu_{ph} (1 - \bar{z}_q) \theta_{pqh}^2 \quad (4.4.45)
\end{aligned}$$

subject to:

$$\begin{aligned}
\sum_{n \in N_a} b f_{jn} \alpha_{na} + \beta_a - \sum_{h \in H} Cap_{ah} \gamma_{jh}^1 - \sum_{h \in H} \widetilde{Cap}_{ah} \gamma_{jh}^2 - \sum_{p \in \Pi(j)} \sum_{h \in H_p} \hat{\mu}_{pha} \sigma_{pjh} \geq -c_{aj}, \\
\forall a \in AT, j \in L \cap CS_a \quad (4.4.46)
\end{aligned}$$

$$\begin{aligned}
\sum_{n \in N_a} b f_{jn} \alpha_{na} - \sum_{h \in H} Cap_{ah} \gamma_{jh}^1 - \sum_{h \in H} \widetilde{Cap}_{ah} \gamma_{jh}^2 - \sum_{p \in \Pi(j)} \sum_{h \in H_p} \hat{\mu}_{pha} \sigma_{pjh} \geq -c_{aj}, \\
\forall a \in AT, j \in L \setminus CS_a \quad (4.4.47)
\end{aligned}$$

$$\sum_{n \in N_a} b g_{gn} \alpha_{na} + \beta_a \geq 0, \quad \forall g \in G_a \cap CS_a, a \in AT \quad (4.4.48)$$

$$\sum_{n \in N_a} b g_{gn} \alpha_{na} \geq 0, \quad \forall g \in G_a \setminus CS_a, a \in AT \quad (4.4.49)$$

$$\sum_{j \in L: q \in \Pi(j)} \gamma_{jh}^1 + \tau_{ph}^1 + \theta_{pqh}^1 \geq f_{qh} - \delta, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph}, q \neq p \quad (4.4.50)$$

$$\begin{aligned}
\sum_{j \in L: p \in \Pi(j)} \gamma_{jh}^1 + \sum_{j \in L: p \in \Pi(j)} \gamma_{jh}^2 + \tau_{ph}^1 + \tau_{ph}^2 + \sum_{j \in L: p \in \Pi(j)} \sigma_{pjh} + \sum_{q \in \Pi_{ph}, q \neq p} \theta_{pqh}^2 \geq f_{ph}, \\
\forall p \in \Pi, h \in H_p : \mu_{ph} > \max_{a \in AT} Cap_{ah} \quad (4.4.51)
\end{aligned}$$

$$\begin{aligned}
\sum_{j \in L: p \in \Pi(j)} \gamma_{jh}^1 + \sum_{j \in L: p \in \Pi(j)} \gamma_{jh}^2 + \tau_{ph}^1 + \sum_{j \in L: p \in \Pi(j)} \sigma_{pjh} + \sum_{q \in \Pi_{ph}, q \neq p} \theta_{pqh}^2 \geq f_{ph}, \\
\forall p \in \Pi, h \in H_p : \mu_{ph} \leq \max_{a \in AT} Cap_{ah} \quad (4.4.52)
\end{aligned}$$

$$\beta_{pqh} \theta_{pqh}^1 - \theta_{pqh}^2 \geq 0, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph}, q \neq p \quad (4.4.53)$$

$$\lambda_k \geq -M, \quad \forall k \in L^M \cup L^O \quad (4.4.54)$$

$$(\beta, \gamma, \tau, \sigma, \theta) \geq 0, \quad (\lambda, \alpha) : \text{unrestricted.} \quad (4.4.55)$$

The optimal dual variable values $(\lambda, \alpha, \beta, \gamma, \tau, \sigma, \theta) \in \Delta$ obtained by solving the subproblem are used to generate the Bender's cut (4.4.21) as follows:

$$\begin{aligned}
\eta_0 \leq & \sum_{p \in \Pi} \left(\sum_{h \in H_p} \mu_{ph} \tau_{ph}^1 \right) z_p + \sum_{\substack{p \in \Pi \\ : \mu_{ph} > \\ \max_{a \in AT} Cap_{ah}}} \left(\sum_{h \in H_p} \tilde{\mu}_{ph} \tau_{ph}^2 \right) z_p \\
& + \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} [\beta_{pqh} \mu_{ph} \theta_{pqh}^1 - \mu_{ph} \theta_{pqh}^2] z_q + \sum_{k \in L^O} \sum_{j \in U_k} \lambda_k \xi_j \\
& + \left[\sum_{k \in L^M} \lambda_k + \sum_{a \in AT} NA_a \beta_a + \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \mu_{ph} \theta_{pqh}^2 \right]. \tag{4.4.56}
\end{aligned}$$

(Step A1.3) Re-optimization with Special Ordered Sets: Once an (integral) optimal solution (z^*, ξ^*) to $\overline{\text{SDFAM}^+}$ is obtained, i.e., profitable optional legs have been selected and the flights have been judiciously retimed, we fix $(z, \xi) = (z^*, \xi^*)$ and simplify the resulting flight network by deleting unnecessary legs. The reduced model SDFAM (which is a fleet assignment model with recapture, denoted **FAMR**) can be written as follows, where $\widehat{\Pi} \subseteq \Pi$ and $\widehat{L} \subseteq L$ denote, respectively, the set of paths and the set of legs resulting from this fixing procedure, and where $\widehat{\Pi}_{ph}$ is defined accordingly (similar to Π_{ph}) based on the activated paths:

$$\text{FAMR : Maximize} \quad \sum_{p \in \widehat{\Pi}} \sum_{h \in H_p} \sum_{q \in \widehat{\Pi}_{ph}} f_{qh} \pi_{pqh} - \sum_{a \in AT} \sum_{j \in \widehat{L}} c_{aj} x_{aj} - \delta \sum_{p \in \widehat{\Pi}} \sum_{h \in H_p} \sum_{q \in \widehat{\Pi}_{ph}, q \neq p} \pi_{pqh} \tag{4.4.57}$$

subject to:

$$\sum_{a \in AT} x_{aj} = 1, \quad \forall j \in \widehat{L} \tag{4.4.58}$$

$$\sum_{j \in \widehat{L}} b f_{jn} x_{aj} + \sum_{g \in G_a} b g_{gn} w_g = 0, \quad \forall n \in N_a, \forall a \in AT \tag{4.4.59}$$

$$\sum_{j \in CS_a \cap \widehat{L}} x_{aj} + \sum_{g \in CS_a \cap G_a} w_g \leq NA_a, \quad \forall a \in AT \tag{4.4.60}$$

$$n_r^l \leq \sum_{a \in AT} \sum_{j \in L_r \cap \widehat{L}} x_{aj} \leq n_r^u, \quad \forall r \in R \tag{4.4.61}$$

$$\sum_{p \in \widehat{\Pi}} \sum_{q \in \widehat{\Pi}_{ph} \cap \widehat{\Pi}(j)} \pi_{pqh} \leq \sum_{a \in AT} Cap_{ah} x_{aj}, \quad \forall j \in \widehat{L}, h \in H \quad (4.4.62)$$

$$\sum_{q \in \widehat{\Pi}_{ph}} \pi_{pqh} \leq \mu_{ph}, \quad \forall p \in \widehat{\Pi}, h \in H_p \quad (4.4.63)$$

$$\pi_{pqh} \leq \beta_{pqh}(\mu_{ph} - \pi_{pph}), \quad \forall p \in \widehat{\Pi}, h \in H_p, q \in \widehat{\Pi}_{ph}, q \neq p \quad (4.4.64)$$

$$x \text{ binary}, (w, \pi) \geq 0. \quad (4.4.65)$$

To solve Problem FAMR, we apply SOS Type I branching based on (4.4.58), as in Jiang and Barnhart (2009). Denoting $Cap_a = \sum_{h \in H} Cap_{ah}$ as the capacity of aircraft type a , after solving the LP-relaxation at any node, we select a leg $j' \in \widehat{L}$ such that $\bar{x}_{aj'}$ is fractional for some set of aircraft types $a \in AT$, where \bar{x} solves $\overline{\text{FAMR}}$, the LP relaxation of FAMR. Among these aircraft types, we extract two indices a_1 and a_2 for the corresponding two largest fractional variables $x_{a_1 j'}$ and $x_{a_2 j'}$ (with $\bar{x}_{a_1 j'} \geq \bar{x}_{a_2 j'}$), and assign these two aircraft types to the sets B_1 and B_2 , respectively. For the remaining aircraft types, assuming that $a \in AT$ are ordered according to nonincreasing Cap_a value and defining $\overline{Cap} = \sum_{a \in AT} Cap_a \bar{x}_{aj'}$, we partition them into $\{a \in AT \setminus \{a_1, a_2\} : Cap_a \geq \overline{Cap}\}$ and $\{a \in AT \setminus \{a_1, a_2\} : Cap_a < \overline{Cap}\}$, and append these two sets to B_1 and B_2 , respectively. Then we branch on the dichotomy: $\{\sum_{a \in B_1} x_{aj'} = 1\} \vee \{\sum_{a \in B_2} x_{aj'} = 1\}$.

Remark 4.1. Several comments related to the application of Benders decomposition are in order at this point. First, note that, in general, whenever the application of the standard Benders decomposition methodology for a particular problem tends to generate more feasibility cuts than optimality cuts, then Saharidis and Ierapetritou (2010) have shown that the overall procedure can greatly benefit by deriving additional cuts based on *maximum feasible subsystems* (MFS), where, for an infeasible problem, an MFS is a maximal subset of constraints that yields feasibility. In this context, whenever a feasibility cut is generated based on the occurrence of an infeasible subproblem, an associated optimality cut can also

be generated by using an MFS of the subproblem. Note that such a cut is valid because it is based on a dual feasible solution to the subproblem in which certain dual variables (corresponding to the constraints of the subproblem that are absent from the derived MFS) are restricted to be zero. In our computation with Model $\overline{\text{SDFAM}}^+$ (see Section 4.5 for details) the percentage of times we encountered infeasible subproblems (with some artificial variable positive) was 3.27%, on average. Hence, considering the low level of occurrences of infeasibility over the total number of Benders iterations, utilizing artificial variables in the subproblem to handle infeasibility issues sufficed in our context. In this same connection, Fischetti et al. (2010) have presented a unifying framework for generating optimality and feasibility cuts by adopting a variable to scale the right-hand side of the dual subproblem and by accommodating a suitable term in the objective function involving this variable along with a normalization constraint. The resulting algorithm automatically generates optimality cuts and feasibility cuts depending on whether this new variable is positive or zero at optimality, and is shown to yield a much more robust performance. Second, although we have adopted a standard implementation of Benders decomposition to investigate its basic utility, there are several algorithmic variants that could be used to further improve the performance of Step A1.2. For example, McDaniel and Devine (1977) have shown that the application of Benders methodology to a decomposable mixed-integer program (MIP) can be accelerated by deriving an initial set of cuts based on solving the linear programming (LP) relaxation to the MIP, before switching over to iteratively solving discrete master programs. Furthermore, Geoffrion and Graves (1974) have proposed an implementation scheme in which the overall master program is essentially solved as a single MIP by progressively generating Benders cuts each time a feasible solution is detected to the current relaxed version of the master program that is better than the incumbent solution at hand. This strategy thus circumvents the expensive task of solving each relaxed master program to optimality before generating a Benders cut. In this context, Cote and Laughton (1984) have demonstrated that such a procedure can further benefit by actively searching for good quality solutions to the current relaxed master program via some appropriate heuristic. However, to assure convergence, the procedure must

ultimately revert to solving the master program exactly as in the approach of Geoffrion and Graves. In a similar vein, Rei et al. (2009) hybridized Benders decomposition with the local branching strategy of Fischetti and Lodi (2003) by using the latter to explore a defined neighborhood of the solution obtained for the master program at each iteration. This process helps detect additional solutions based on which multiple optimality or feasibility cuts (or combinatorial cuts that eliminate specific solutions) can be generated. Computational results pertaining to two classes of network design problems demonstrate that such an approach can greatly improve the performance of Benders algorithm. Furthermore, Magnanti and Wong (1981) have shown that whenever the dual subproblem has alternative optimal solutions, the generated Benders cut may not necessarily be effective in the sense that it might be possible to uniformly dominate this cut with another valid Benders cut. They therefore solve an auxiliary dual subproblem with a modified objective function to derive an additional non-dominated or Pareto optimal Benders cut. Computational results on some network design problems indicate that this strategy significantly accelerates the convergence of the Benders decomposition algorithm. Recently, Sherali and Lunday (2010) have demonstrated that a class of nondominated Benders cuts can be directly generated by suitably perturbing the original dual subproblem objective function (or the right-hand side of the primal subproblem) in a manner that essentially renders the fixed complicating variable vector used to formulate the subproblem fully dense. The authors exhibit the computational benefits of their strategy using several specially structured as well as generic test problems. Another novel idea in this connection has been proposed by Saharidis et al. (2010), where multiple valid Benders cuts are generated at each iteration that progressively involve additional complicating variables. The motivation for this strategy, called the Covering Cut Bundle (CCB) method, is that standard Benders cuts often tend to be sparse, and therefore, several such cuts are required to sufficiently tighten the relaxed master program. By generating a bundle of gradually increased density Benders cuts (implemented a maximum number of times per iteration or until all complicated variables are sufficiently represented in the derived set of cuts), the authors exhibit a significant speedup for two scheduling applications due to a reduced number

of Benders iterations, leading to the solution of fewer discrete relaxed master programs. For our model $\overline{\text{SDFAM}}^+$, the average density of Benders cuts in our test results turns out to be 24.3% (see Section 4.5 for details). Hence, in a more refined algorithmic approach, it might be beneficial to implement the CCB method as recommended by Saharidis et al. (2010). Our results in the following section indicate that even a straightforward application of Benders decomposition at Step A1.2 greatly benefits the algorithmic process. A further enhancement of the proposed algorithm is possible by implementing several of the foregoing strategies, and we recommend a more detailed investigation of such techniques for future research. \square

Algorithm A2:

For Algorithm A2, we adopt the same approach as in Algorithm A1, except that at Step A1.2, we solve the relaxed MIP given by Model $\overline{\text{SDFAM}}^+$ directly (using CPLEX), instead of applying Benders decomposition to this problem.

4.5 Computational Experiments

Two sets of computational experiments are performed in this section. In Section 4.5.1, as a preliminary step to identify effective algorithmic strategies, we use the different algorithms described in Section 4.4 to solve the *baseline model* where the constraints corresponding to the additional modeling features incorporated within SDFAM are dropped (in essence, this is the model presented in Chapter 3). Then, in Section 4.5.2, we augment the baseline model by individually accommodating the features of flexible schedules, schedule balance, and recapture one at a time, as well as solve SDFAM where all of these features are simultaneously included, and we analyze the impact of these additional considerations.

For experimental purposes, we designed five realistic flight networks based on real data provided by *United Airlines* as follows: N1 (466 flights, 2121 paths), N2 (592 flights, 3342 paths), N3 (712 flights, 4649 paths), N4 (848 flights, 6880 paths), and N5 (1016 flights, 10726

paths). On average, the proportion of non-stop, two-leg, and three-leg itineraries were about 15.4%, 73.1%, and 11.5%, respectively. All runs were made using the commercial software OPL CPLEX 10.1 on a Precision T5500 computer having an Intel Xeon 2.13 GHz processor, with 4.00 GB of RAM, and running Windows 7. For the purpose of comparison, we present in the reported tables in Sections 4.5.1 and 4.5.2 the quality of the solution attained in relative terms. More specifically, in the comparison of the algorithmic strategies in Section 4.5.1, **% gap** is defined as $100(F^* - F)/F^*$, where F denotes the objective value attained by the particular method, and F^* denotes the best (maximum) objective value found across all procedures. Furthermore, when analyzing the different modeling features in Section 4.5.2, **% Improvement** refers to the percentage improvement in profits obtained over the baseline case for each modeling variation, and is defined as $100(G - G_0)/G_0$, where G_0 denotes the objective value obtained for the baseline case, and G denotes the objective value attained for the particular variant. For reporting CPU times, we provide the actual CPU effort in hours. The run-time limit was set as 12 CPU hours, except for the last more elaborate experiment where we set this limit as 24 CPU hours. A default optimality tolerance of $\epsilon = 10^{-6}$ was used for all runs.

4.5.1 Effectiveness of Algorithmic Solution Approaches

In this section, we explore the utility of the valid inequalities derived from our polyhedral analysis, as well as that of the Benders decomposition approach, by solving the baseline model using Algorithms A0, A1, and A2, where A0 represents a direct solution of the model using the CPLEX solver without any of the aforementioned enhancements, and where Algorithms A1 and A2 are described in Section 4.4. The Benders iterative process within Algorithm A1 was set to generate a maximum of 100 Benders cuts. This limit was selected based on some preliminary tests where we observed that there was not much improvement in the objective function value (less than 0.01%) between 80 to 100 Benders iterations, and that generating additional cuts became computationally expensive while providing marginal (if

at all) improvements in solution quality. Therefore, to be consistent across all cases, we imposed a limit of 100 Benders cuts.

Table 4.1: Comparative Results for Algorithms A0, A1, and A2

| Network | Algorithms | | | | |
|-----------------|------------------|-------------------|-------------|-------------|-------------|
| | | | A0 | A1 | A2 |
| N1: | LP-IP gap | % gap | 5.15 | 0.00 | 0.00 |
| | 7.3% | CPU (hrs.) | 12.0 | 6.5 | 9.8 |
| N2: | LP-IP gap | % gap | 6.08 | 0.00 | 1.43 |
| | 8.1% | CPU (hrs.) | 12.0 | 7.3 | 12.0 |
| N3: | LP-IP gap | % gap | 5.79 | 0.00 | 1.64 |
| | 7.8% | CPU (hrs.) | 12.0 | 9.3 | 12.0 |
| N4: | LP-IP gap | % gap | 7.10 | 0.00 | 1.78 |
| | 6.6% | CPU (hrs.) | 12.0 | 10.2 | 12.0 |
| N5: | LP-IP gap | % gap | 7.95 | 0.00 | 1.03 |
| | 5.7% | CPU (hrs.) | 12.0 | 12.0 | 12.0 |
| Average: | | | | | |
| | LP-IP gap | % gap | 6.41 | 0.00 | 1.18 |
| | 7.1% | CPU (hrs.) | 12.0 | 9.1 | 11.6 |

The results in Table 4.1 indicate that, for all cases, the best objective function value was obtained and the least CPU time was consumed when we used Algorithm A1, where both valid inequalities and Benders decomposition are applied. The average % gaps realized by A0 and A2 relative to A1 are 6.41% and 1.18%, respectively. On average, about 21% of optional legs were found to be non-profitable and, hence, dropped. The LP-IP gap at the root node for the baseline model (with respect to the best solution detected) was 7.3%,

8.1%, 7.8%, 6.6%, and 5.7% for the instances N1, N2, N3, N4, and N5, respectively. The CPU time required by Algorithm A0 exceeded the 12 hour limit in all instances, whereas the average CPU time consumed by Algorithm A2 was 11.6 hours (4% improvement), which demonstrates the tightening effect of the proposed valid inequalities added in Algorithm A2. The average CPU time required by Algorithm A1 was 9.1 hours, which amounts to a further reduction by 21% of the effort in comparison to Algorithm A2, thus exhibiting the utility of applying Benders decomposition.

Table 4.2: Performance Characteristics of Algorithm A1

| Network | N1 | N2 | N3 | N4 | N5 | Average |
|---------------------------------|--------|------|--------|------|------|------------------|
| Number of Benders cuts (iters.) | 82 | 100 | 94 | 100 | 100 | 95.2 |
| Time spent in CPU hrs.: | | | | | | |
| Step A1.1 | 0.39 | 0.29 | 0.36 | 0.49 | 0.44 | 0.4 |
| Step A1.2 | 4.19 | 5.03 | 6.18 | 6.90 | 7.76 | 6.0 |
| Step A1.3 | 1.92 | 1.98 | 2.96 | 2.81 | 3.80 | 2.7 |
| % Feasibility cuts* | 1.2(1) | 2(1) | 2.1(2) | 7(2) | 4(2) | 3.27(1.7) |
| % Density of cuts | 20.3 | 16.5 | 31.9 | 28.8 | 24.2 | 24.3 |

* The average number of positive artificial variables at optimality for infeasible subproblems are provided in parenthesis.

To further elucidate the performance characteristics of Algorithm A1, Table 4.2 provides the following additional statistics: (i) the number of cuts generated (Benders iterations); (ii) a breakdown of the time spent in each of the steps A1.1, A1.2, and A1.3; (iii) the percentage of cuts that were feasibility cuts, and the number of artificial variables that were positive on average over the cases where feasibility cuts were generated; and (iv) the average percentage density of the generated Benders cuts. An average of 95.2 Benders cuts needed to be generated over the test cases, with the instances N2, N4, and N5 requiring the maximum limit of 100 cuts. On average, Steps A1.1, A1.2, and A1.3 used 4.4%, 66.2%, and 29.4% of the run-times, respectively, which indicates that almost two-thirds of the total effort was consumed at Step A1.2. For the most part, the algorithm generated optimality cuts, with the average percentage of feasibility cuts being just 3.27%, and where only one to two artificial variables were positive in the subproblem solution whenever a feasibility

cut was generated. Finally, the average density of the derived Benders cuts in terms of the percentage of complicating variables covered in each cut was about 24.3%, which is moderately low. Hence, Algorithm A1 might further benefit by the implementation of the CCB method of Saharidis et al. (2010) as previously discussed in Remark 4.1, which we recommend for future research.

4.5.2 Modeling Features

In this section, we assess the separate effects of incorporating the features of flexible schedules, schedule balance, and recapture, by using each of these individually in turn to augment the baseline model (which includes none of them). We solve the resulting problems based on these three cases, respectively, and analyze the results obtained. Thereafter, we incorporate all of the aforementioned additional features as presented in Model SDFAM and compare the results obtained with those of the baseline model.

Effect of Flexible Flight Schedules

For each network dataset, we generated three additional test instances pertaining to flight retiming windows of durations 10(5), 20(5), and 20(10) minutes, where the corresponding time interval between leg copies is specified in parenthesis. Hence, for example, flight time-window 20(5) permits leg departure time deviations of $\{-10, -5, 0, +5, +10\}$ minutes from the original departure time. For each of these cases, feasible paths or itineraries were generated by constructing different combinations of possible leg connections. In order to render the scenarios more practical, we let the corresponding itinerary demands depend on the available connection times for passengers. Considering that flight retiming among the multiple copies of a leg on any given path has an effect on the demand distribution of that path, we set the ideal passenger connection time as 90 minutes for the sake of illustration (assuming a domestic regional subproblem), and we assumed in the data generation process that the

further the passenger connection time is from 90 minutes, the corresponding itinerary attracts less demand in reference to the demand with respect to the original itinerary. Also, we excluded from consideration any resulting itineraries based on flight retimings whenever the connection times fell below 60 minutes or exceeded 120 minutes. We note here that minimum connection times vary by stations, and the relationship between connection times and demand is market dependent and is influenced by other consumer preference factors. Hence, our settings are an approximation used only for illustrative purposes, and airlines would need to conduct a more detailed sensitivity analysis in this vein.

Table 4.3 presents the results obtained. In every data instance (N1-N5), the model with flight retiming performed better than the model with fixed departure times (the baseline case), producing more profitable solutions by 1.85%, 2.67%, and 3.69%, on average for the time-window cases 10(5), 20(5), and 20(10), respectively. For the data instance N1, all the runs with different time-window settings were completed within 12 CPU hours, where the time-window case 20(5) performed the best (5.57% improvement in profits over the baseline case) while consuming the most computational time (about 11.2 CPU hours). However, with an increase in the size of data instances, the time-window 20(10) uniformly outperformed the other two cases, thereby demonstrating the advantage of having the leg copies spaced out both reasonably and manageably. Although all the runs with multiple leg copies terminated at the 12 CPU hour time limit for the larger test instances N3-N5 (with the case 20(5) having insufficient memory for the instance N5), yet the case 20(10) improved profits by 3.52%, 4.25%, and 3.85% on average for the instances N3, N4, and N5, respectively, as compared with the baseline case.

From the standpoint of schedule planning and airline operations, it is of interest to measure the proportion of flights that are actually retimed from the originally scheduled time when compared with the baseline case. The statistic *% Retimed Legs* provides this information. Both the network size (as predicated by the number of flight legs) and the number and timings of available leg copies affect the nature and quality of the generated schedules within the set time limit. For the relatively smaller instances N1 and N2, more legs

Table 4.3: Comparative Results for Different Flight Retiming Windows

| Network | Retiming Windows | | | |
|-----------------------|------------------|-------------|--------------|-------------|
| | 0 | 10(5) | 20(5) | 20(10) |
| N1: | | | | |
| % Improvement | 0.00 | 0.96 | 5.57 | 2.18 |
| CPU (hrs.) | 6.5 | 9.7 | 11.2 | 9.9 |
| % Retimed Legs | N/A | 16.7 | 22.5 | 20.6 |
| N2: | | | | |
| % Improvement | 0.00 | 2.72 | 3.81 | 4.64 |
| CPU (hrs.) | 7.3 | 11.0 | 12.0 | 11.4 |
| % Retimed Legs | N/A | 18.1 | 21.7 | 21.0 |
| N3: | | | | |
| % Improvement | 0.00 | 1.81 | 0.96 | 3.52 |
| CPU (hrs.) | 9.3 | 12.0 | 12.0 | 12.0 |
| % Retimed Legs | N/A | 20.4 | 19.9 | 22.8 |
| N4: | | | | |
| % Improvement | 0.00 | 1.88 | 0.36 | 4.25 |
| CPU (hrs.) | 10.2 | 12.0 | 12.0 | 12.0 |
| % Retimed Legs | N/A | 20.7 | 19.6 | 21.8 |
| N5: | | | | |
| % Improvement | 0.00 | 1.90 | insufficient | 3.85 |
| CPU (hrs.) | 12.0 | 12.0 | memory | 12.0 |
| % Retimed Legs | N/A | 19.5 | | 21.6 |
| Average: | | | | |
| % Improvement | 0.00 | 1.85 | 2.67 | 3.69 |
| CPU (hrs.) | 9.1 | 11.4 | 11.8 | 11.5 |
| % Retimed Legs | N/A | 19.1 | 20.9 | 21.5 |

were retimed for the time-window case 20(5) for which there exist relatively more leg copies to select from than for the other two cases. However, this effect diminished as the problem size increased because of the accompanying combinatorial burden of composing profitable

sequences of flights for different itineraries. On average, the proportion of the number of retimed flight legs were 19.1%, 20.9%, and 21.5% for the time-window cases 10(5), 20(5), and 20(10), respectively. The number of such resulting retimed flight legs were affected by factors such as the connection time between consecutive legs in the different itineraries, the maximum allowable (departure) time shift for each leg copy, and the connection times that fall within the range of 60 minutes and 120 minutes after possible combinations of leg copies are considered. As a point of interest, the average passenger connection times for the different time-window configurations of 0, 10(5), 20(5), and 20(10) turned out to be 103.27, 100.57, 99.14, and 97.38 minutes, evidently tending toward the assumed ideal connection time of 90 minutes based on the demand specifications.

Overall, within the 12 CPU hours time limit, the time-window case of 20(10) performed better than the other two retiming cases. For this case, where flight times were permitted to change by ± 10 minutes, the estimated increase in profit was about \$41,000/day (\$14.9M/year) over the baseline case, whereas when flight times were allowed to vary by up to ± 5 minutes, the estimated increase in profit was about \$18,000/day (\$6.5M/year) over the baseline case. Having a relatively smaller time interval between leg copies significantly increases the problem size, and the resulting burden consequently inhibited the latter case from achieving better objective function values over the former case, particularly due to the fact that there were evidently insufficient incentives for flights to depart only five minutes away from their scheduled times, as compared with using deviations of ten minutes.

Effect of Schedule Balance

For this experiment, two factors were considered in generating a set of flight groups, namely, the stations at which to impose the schedule balance constraint (4.2.9), and the number of groups to consider at each station along with the associated time intervals. Due to the hub-and-spoke structure of typical major carrier networks, some stations might have only one pair of flights per day (i.e., one inbound and one outbound), while other stations might have

optional legs under consideration during morning and evening hours, where no more than one leg can be selected in each of these time ranges, and yet others such as busy major hubs might have more than 100 flight operations per day. For modeling purposes, we generated a group r by first selecting stations of relevant interest, and for each station under consideration, we specified certain appropriate daily time blocks; for example, for a large congested hub, the time blocks were specified as 6am - 10am, 10:01am - 4pm, 4:01pm - 8pm, and 8:01pm - midnight. Considering that the purpose of including schedule balance in our model is to capture adequate market shares, we composed the sets L_r by using certain pertinent flight legs within each time block at a given station, and then accordingly specified the minimum (n_r^l) and maximum (n_r^u) number of flights to be selected in each such designated group $r \in R$.

For comparison purposes, we developed two different sets of flight groups to measure model performance under different scenarios, namely, $Balance_1$ and $Balance_2$. In the $Balance_1$ scenario, we selected nine major stations where there exist more than 40 daily flight operations per station (i.e., departures and arrivals), while in the $Balance_2$ scenario, we selected 14 major stations where there exist more than 30 daily flight operations per station. Based on the given data sets, on average, 32% and 65% of all the itineraries involve at least one of these stations in the two respective scenarios. Naturally, imposing the additional restrictions of schedule balance will diminish the profits represented in the objective function for the given data instance, but this reduction needs to be balanced with the potential improvement in customer goodwill and an accompanying increase in market share. Hence, for each schedule balance scenario, we also modified the data to reflect a 15% increase in demand on the affected itineraries that involve the balanced stations. These corresponding enhanced demand scenarios are designated as $Balance_i^+$ for $i = 1, 2$. Note that a 15% increase in demand on affected itineraries is used here only for illustrative purposes, and is rather optimistic. In practice, this would require a market analysis. Moreover, an airline can perform a sensitivity analysis on this value to assess the extent to which schedule balance should augment demand to make this strategy beneficial.

Table 4.4 presents the results obtained. For the simply more restricted schedule balance

Table 4.4: Computational Results with Schedule Balance Consideration

| Network | Baseline | Balance ₁ | Balance ₂ | Balance ₁ ⁺ | Balance ₂ ⁺ |
|----------------------|-------------|----------------------|----------------------|-----------------------------------|-----------------------------------|
| N1: | | | | | |
| % Improvement | 0.00 | -2.47 | -3.12 | 1.54 | 0.96 |
| CPU (hrs.) | 6.5 | 7.2 | 7.3 | 6.8 | 7.1 |
| N2: | | | | | |
| % Improvement | 0.00 | -2.58 | -3.34 | 0.85 | 0.25 |
| CPU (hrs.) | 7.3 | 7.7 | 7.9 | 7.9 | 8.4 |
| N3: | | | | | |
| % Improvement | 0.00 | -1.73 | -3.90 | 2.17 | 0.46 |
| CPU (hrs.) | 9.3 | 9.4 | 9.5 | 9.5 | 9.7 |
| N4: | | | | | |
| % Improvement | 0.00 | -1.90 | -2.81 | 1.04 | 0.23 |
| CPU (hrs.) | 10.2 | 10.3 | 10.8 | 10.2 | 10.7 |
| N5: | | | | | |
| % Improvement | 0.00 | -2.04 | -3.11 | 1.98 | 0.53 |
| CPU (hrs.) | 12.0 | 12.0 | 12.0 | 12.0 | 12.0 |
| Average: | | | | | |
| % Improvement | 0.00 | -2.14 | -3.26 | 1.52 | 0.49 |
| CPU (hrs.) | 9.1 | 9.4 | 9.6 | 9.3 | 9.7 |

scenarios, the reduction in profit as compared with the baseline case was 2.14 % and 3.26 % on average for Balance₁ and Balance₂, respectively. Furthermore, when compared with the baseline model, accommodating Balance₁ consumed about 0.3 more CPU hours on average, whereas the model accommodating Balance₂ required an additional CPU hours of about 0.5 on average.

On the other hand, when we account for the aforementioned enhancement in market shares when accommodating schedule balance, the situation reverses. In this case, Balance₁⁺ and Balance₂⁺ achieved higher profits by 1.52% and 0.49%, respectively, as compared with the baseline case. Evidently, the increase in demand on the itineraries of interest more than

compensates for the accompanying schedule restrictions, thereby contributing to greater profits. As mentioned above, airlines therefore need to carefully assess the effect of schedule balance on demand while additionally considering other intangible factors such as goodwill and loyalty. Moreover, it is of interest to note that in the Balance_1^+ and Balance_2^+ scenarios as compared with the Balance_1 and Balance_2 scenarios, larger aircraft types, if available, tend to be assigned to pertinent flight segments where the capacities are already saturated. If no such larger aircraft exist to match increased demands, some passengers might be spilled depending on the shortage between capacity and demand. In this situation, the possibility of recapturing a portion of the spilled demand becomes relevant, as explored in the following section.

Recapture Effect

For this experiment, we delineated a set of OD markets with respect to which compatible itineraries are identified for recapture consideration. Accordingly, we generated the recapture rates, β_{pqh} , for such compatible itineraries by extracting the QSI data for each relevant itinerary and using the formula given in Kniker (1998). We also experimented with different re-booking or re-routing penalty parameter values $\delta \in \{1, 50, 100, 200\}$ (dollars per passenger, abbreviated \$/PAX), to assess the sensitivity of fleeting decisions to the relative desire of routing passengers on their original itinerary choices. In addition, for each δ -value, we assessed the proportion of demand that is re-accommodated on alternative compatible routes in the resulting solution by computing $100(\sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \pi_{pqh}) / (\sum_{p \in \Pi} \sum_{h \in H_p} \mu_{ph})$; this is denoted as *% Recap demand* in Table 4.5.

Table 4.5 clearly demonstrates that being able to recapture spilled demand and accommodate it on itineraries where seats are available on all constituent flight legs achieves greater profits, but as the penalty parameter δ increases, this improvement in profits diminishes due to fewer recapture occurrences. On average, the respective cases of $\delta = 1, 50, 100,$ and 200 , improved profits over the baseline case by 2.10%, 2.09%, 2.02%, and 1.92%, which

Table 4.5: Computational Results with Recapture Consideration

| Network | Baseline | δ (\$/PAX) | | | |
|-----------------------|-------------|-------------------|-------------|-------------|-------------|
| | | 1 | 50 | 100 | 200 |
| N1: | | | | | |
| % Improvement | 0.00 | 2.41 | 2.39 | 2.28 | 2.16 |
| CPU (hrs.) | 6.5 | 8.9 | 8.8 | 8.5 | 8.2 |
| % Recap demand | N/A | 5.0 | 4.8 | 4.5 | 4.1 |
| N2: | | | | | |
| % Improvement | 0.00 | 2.93 | 2.91 | 2.79 | 2.67 |
| CPU (hrs.) | 7.3 | 9.7 | 9.7 | 9.6 | 9.5 |
| % Recap demand | N/A | 5.7 | 5.6 | 5.3 | 5.0 |
| N3: | | | | | |
| % Improvement | 0.00 | 2.08 | 2.07 | 2.01 | 1.92 |
| CPU (hrs.) | 9.3 | 10.9 | 10.8 | 10.6 | 10.7 |
| % Recap demand | N/A | 5.2 | 5.1 | 4.8 | 4.6 |
| N4: | | | | | |
| % Improvement | 0.00 | 1.23 | 1.23 | 1.17 | 1.09 |
| CPU (hrs.) | 10.2 | 12.0 | 12.0 | 12.0 | 12.0 |
| % Recap demand | N/A | 4.1 | 4.1 | 3.9 | 3.7 |
| N5: | | | | | |
| % Improvement | 0.00 | 1.86 | 1.85 | 1.83 | 1.77 |
| CPU (hrs.) | 12.0 | 12.0 | 12.0 | 12.0 | 12.0 |
| % Recap demand | N/A | 3.9 | 3.9 | 3.7 | 3.4 |
| Average: | | | | | |
| % Improvement | 0.00 | 2.10 | 2.09 | 2.02 | 1.92 |
| CPU (hrs.) | 9.1 | 10.7 | 10.7 | 10.5 | 10.4 |
| % Recap demand | N/A | 4.8 | 4.7 | 4.4 | 4.1 |

translates to estimated increases in annual profits of \$24.6M, \$24.5M, \$23.7M, and \$22.5M. The accompanying computational effort for each case of $\delta = 1, 50, 100,$ and 200 was 10.7, 10.7, 10.5, and 10.4 hours, respectively, showing a slight decrease as δ increases because of

fewer occurrences of switched itineraries. The average proportions of recaptured demands for the cases with $\delta = 1, 50, 100,$ and 200 were 4.8%, 4.7%, 4.4%, and 4.1%, respectively, displaying a limiting marginal rate of recapture as the penalty parameter decreases close to 0. Furthermore, a key performance measure for airlines is the average load factor, given by the average across all flights of the ratio of the number of actual passengers served (from all incident itineraries) to the capacity of the aircraft type assigned to the particular leg segment. The average load factor for the baseline case turns out to be 76.3%, whereas that for the cases with $\delta = 1, 50, 100,$ and 200 was computed as 83.0%, 82.8%, 81.9%, and 81.2%, respectively, which reflects a better matching of capacity with demand. Additionally, as a point of interest, the average percentage of demand being served across all itineraries was 90.6%, 90.3%, 89.1%, and 87.4% corresponding to $\delta = 1, 50, 100,$ and $200,$ respectively.

Impact of the Overall Integrated Model

In this section, we examine the proposed Model SDFAM, where all of the foregoing additional features of flight retiming, schedule balance, and recapture are integrated concurrently, and we provide comparisons with the baseline model to assess the overall effect on solution quality and computational effort using Algorithm A1. Note that, in Chapter 3, we have provided empirical evidence to amply demonstrate that the baseline model itself potentially yields an estimated \$13.7M - \$28.3M increase in annual profits in comparison with a sequential schedule design and fleet assignment approach, even while considering two different look-ahead intensity levels within the sequential approach. Hence, we do not repeat this particular comparative analysis here. Also, because of the additional complexity of Model SDFAM, we extended the run-time limit to 24 CPU hours.

For this experiment, we used the following settings in SDFAM_a: (i) the time-window case of 20(10) for defining flexible leg copies; (ii) the Balance₁⁺ scenario for considering schedule balance; and (iii) the median penalty parameter value $\delta=100$ for recapture. In SDFAM_b, as compared with SDFAM_a, we utilized (i) the same time-window setting of 20(10); (ii)

Table 4.6: Comparative Results for the Baseline vs. the Overall Integrated Model

| Network | Baseline | SDFAM _a | SDFAM _b |
|----------------------|-------------|--------------------|--------------------|
| N1 | | | |
| % Improvement | 0.00 | 4.48 | 4.31 |
| CPU (hrs.) | 6.5 | 15.1 | 14.4 |
| N2 | | | |
| % Improvement | 0.00 | 6.32 | 5.88 |
| CPU (hrs.) | 7.3 | 17.0 | 16.6 |
| N3 | | | |
| % Improvement | 0.00 | 6.51 | 6.32 |
| CPU (hrs.) | 9.3 | 18.9 | 18.0 |
| N4 | | | |
| % Improvement | 0.00 | 5.58 | 5.25 |
| CPU (hrs.) | 10.2 | 21.6 | 20.9 |
| N5 | | | |
| % Improvement | 0.00 | out-of- | 4.23 |
| CPU (hrs.) | 13.9 | memory | 24.0 |
| Average: | | | |
| % Improvement | 0.00 | 5.72 | 5.20 |
| CPU (hrs.) | 9.4 | 18.2 | 19.0 |

three busiest hub stations for incorporating schedule balance; and *(iii)* the largest 70% (by itinerary frequency) of OD markets for recapture consideration with $\delta=100$. Overall, the objective function values for SDFAM_a and SDFAM_b improved across all instances by 5.72% and 5.20% on average, respectively, when compared with the baseline case (except that Instance N5 ran out of memory for SDFAM_a, which underscores the need for more sophisticated algorithms (see Remark 4.1 in Section 4.4) as well as more powerful computer hardware for handling such large-scale problems). This improvement amounts to an average increase in annual profits of \$35.1M and \$31.8M, respectively, reflecting the beneficial compounding effects of the retiming, schedule balance, and recapture features incorporated in SDFAM.

The computational effort required to solve SDFAM_a and SDFAM_b were 18.2 and 19.0 CPU hours on average, respectively (18.2 and 17.3 hrs. considering instances N1-N4 that both models solved), which are about twice that consumed for the baseline case. However, given the strategic planning nature of SDFAM, this is acceptable in practice where the emphasis is on the potential improvement in profits.

Chapter 5

An Integrated Schedule Design, Fleet Assignment, and Aircraft Routing Model

5.1 Introduction

In this chapter, we integrate schedule design, fleet assignment, and aircraft routing decisions, while directly incorporating optional legs, itinerary-based demands, multiple fare-classes, flexible flight retimings, maintenance considerations, demand recapture, and through-flights for achieving greater revenue opportunities. A novel flight network representation paradigm is utilized to provide greater flexibility in accommodating integrated operational considerations. More specifically, this model adopts a *flight network* (**FN**) representation (also known as a *connection network* or an *activity-on-node network*), rather than the *time-space network* (**TSN**) (or the *time-line* or *activity-on-edge network*) construct that is traditionally utilized to model airline operational problems (see Clausen et al. 2010). This network representation is similar to the one described by Desaulniers et al. (1997), but exhibits two differences.

First, Desaulniers et al. construct a separate network for each aircraft type (this is also the case for the TSN). Second, they use two different sets of nodes representing the initial and final stations, respectively. Interestingly, it appears that the FN representation offers some distinctive advantages over the TSN. In particular, as Rushmeier and Kontogiorgis (1997) point out, the TSN is not able to distinguish among specific aircraft, which limits its application in the subsequent routing problem. On the other hand, the FN method directly provides routings for individual aircraft, which permits modeling flexible flight times, through-flights, and maintenance requirements among other features, as done herein. In particular, airline companies actively seek to schedule through-flights whenever possible because they generate more revenues as passengers prefer such flights due to their convenience and the assurance of obviating missed connections and lost baggage. Moreover, even in the absence of any routing considerations, the FN might be preferred over the TSN because its representation has a much simpler structure and yields a graph having fewer nodes and arcs. Indeed, in the TSN, with each leg, and for each aircraft type, there are associated two nodes representing the departure and the arrival events, respectively, whereas in the FN, a single node is associated with each flight leg. Furthermore, the FN representation avoids the discretized sets of duplicate arcs that arise by virtue of incorporating flexible schedules within TSN-based models. In order to tighten the model representation and reduce its complexity, we apply the Reformulation-Linearization Technique (RLT) of Sherali and Adams (1990, 1994), and introduce suitable hierarchical symmetry-breaking constraints (see Sherali and Smith, 2001) along with other classes of valid inequalities to enhance the model solvability. For the resulting large-scale augmented model formulation, we design a Benders decomposition-based solution method, and present computational results to demonstrate the efficacy of the proposed approach.

The remainder of this chapter is organized as follows. In Section 5.2, we begin by presenting our proposed integrated schedule design, fleet assignment, and aircraft routing mixed-integer programming model, and then augment the basic integrated model further to introduce a set of symmetry-breaking constraints, as well as to facilitate maintenance-feasible decisions.

Section 5.3 develops algorithmic solution approaches that generate various classes of valid inequalities to tighten the model representation and then utilize Benders decomposition to efficiently optimize the resulting large-scale formulation. Finally, results from our computational experiments are reported and analyzed in Section 5.4.

5.2 Model Description and Notation

In this section, we first present our notation along with the network description, followed by a basic model formulation that is predicated on a flight network representation. We next extend the developed model by incorporating symmetry-breaking constraints to curtail enumeration and thereby accelerate the solution process. Finally, we introduce certain additional aggregate restrictions to promote maintenance-feasible decisions.

Below we present our notation along with the network description, followed by the basic model formulation.

Sets:

F : set of aircraft, indexed by i .

AT : set of aircraft types, indexed by a .

F_a : set of aircraft of type $a \in AT$.

S : set of stations, indexed by s .

F^s : set of aircraft whose initial station is $s \in S$.

F_a^s : set of aircraft of type $a \in AT$ whose initial station is $s \in S$.

$L \equiv \{1, \dots, n\}$: set of flight legs in the flight schedule, indexed by j or k .

$L^M \subseteq L$: set of mandatory legs.

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

L^{TF} : set of designated mandatory through-flight pairs of legs, indexed by (j, k) , with $\{j, k\} \subseteq L$.

A_{TF} : set of pairs of flight legs that can be beneficially used as through-flights if possible, indexed by (j, k) , with $\{j, k\} \subseteq L$.

$F(j)$: set of aircraft $i \in F$ that can possibly serve leg $j \in L$.

$J(i)$: set of legs $j \in L$ that can be served by aircraft i (i.e., for which $i \in F(j)$), $\forall i \in F$.

$L^{a,s}$: common set of legs $j \in L$ that can be possibly served by aircraft $i \in F_a^s$, $\forall a \in AT$, $s \in S$ (i.e., $J(i) \equiv L^{a,s}$, $\forall i \in F_a^s$, for each $a \in AT$, $s \in S$).

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

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

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

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

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

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

$H_p \subseteq H$: set of all fare-classes on path p , $\forall p \in \Pi$.

Π_{ph} : set of paths to which demand in fare-class $h \in H_p$ on path $p \in \Pi$ can be routed (note that $p \in \Pi_{ph}$; and that $q \in \Pi_{ph} \Rightarrow h \in H_q$).

Remark 5.1. Here, $F(j)$, $\forall j \in L$, is a subset of F that is based on the (demand) characteristics of the flight leg segment j , and the capacities of aircraft that can possibly and usefully be assigned to it. Note that, as in Barnhart et al. (2009), different aircraft $i \in F$ can be localized to subnetworks and hence omitted from inclusion within $F(j)$ for legs $j \in L$ lying outside such subnetworks. Indeed, flight legs differ significantly in terms of their corresponding demands, and it would be futile to assign large aircraft to relatively small demand legs, where in particular, such an assignment would be impermissible if either

of the end-node stations of the leg cannot accommodate such large sized aircraft. More specifically, suppose that we classify aircraft whose total capacity is greater than 200 as *large aircraft* (typically wide-body jets that serve relatively long-haul trips between hubs). If a flight leg j is *unconstrained*, that is, its maximum attainable demand is less than the smallest aircraft capacity that can be assigned to it, then we can logically preclude any large aircraft from being assigned to leg j . Such reductions can greatly reduce the problem size and the accompanying computational effort, particularly noting that a significant proportion of flights can be classified as unconstrained given the large variability in demand levels by seasons and locations across the hub-and-spoke flight networks of major airlines. For convenience in notation and implementation, we have also defined by way of an inverse to the sets $F(j), \forall j \in L$, the sets $J(i), \forall i \in F$, which represent the legs that can be possibly served by each aircraft $i \in F$. \square

Parameters:

c_{ij} : cost of assigning aircraft i to leg $j, \forall i \in F, j \in L$.

v_{ijk} : additional profit obtained when the inbound flight j and outbound flight k are flown by the same aircraft $i, \forall (j, k) \in A_{TF}, i \in F(j) \cap F(k)$.

t_{ij} : flying duration of leg j when it is assigned to aircraft $i, \forall i \in F, j \in L$.

τ_i : turn-time of aircraft $i, \forall i \in F$.

$[a_j, b_j]$: departure time-window of leg $j, \forall j \in L$.

Cap_{ih} : capacity of aircraft i to accommodate passengers for fare-class $h, \forall i \in F, h \in H$.

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 $p, \forall p \in \Pi, h \in H_p$.

β_{pqh} : maximum proportion of spilled demand in fare-class $h \in H_p \cap H_q$ on path $p \in \Pi$ that can be possibly routed to a path $q \in \Pi, q \neq p$ (note that $\Pi_{ph} \equiv \{p\} \cup \{q \in \Pi, q \neq p : h \in H_q$

and $\beta_{pqh} > 0\}$, $\forall p \in \Pi, h \in H_p$).

δ : penalty or re-booking cost per passenger recaptured (or re-routed) on an alternative path.

Underlying Network:

We consider the digraph $G = (V, A)$ where the set of nodes V and the set of arcs A are defined as follows. Each leg $j \in L$ is represented by a node j ($j = 1, \dots, n$). Furthermore, nodes $n + s$ ($s \in S$) are added to represent the different stations. The set of arcs A contains three types of arcs:

- $(n + s, j) \in A \Leftrightarrow$ station s ($s \in S$) is the departure station of flight $j \in L$;
- $(j, n + s) \in A \Leftrightarrow$ station s ($s \in S$) is the arrival station of flight $j \in L$;
- $(j, k) \in A \Leftrightarrow$ an aircraft $i \in F(j) \cap F(k)$ can cover leg k immediately after leg j ($j \neq k \in L$). Note that this requires that the departure station of leg k and the arrival station of leg j are the same, and also that the time-window constraints permit this sequence (i.e., $a_j + \min_{i \in F(j) \cap F(k)}(t_{ij} + \tau_i) \leq b_k$).

Decision Variables:

$$x_{ij} : \begin{cases} 1, & \text{if aircraft } i \text{ covers leg } j, \forall j \in L, i \in F(j) \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{ijk} : \begin{cases} 1, & \text{if aircraft } i \text{ serves leg } k \text{ immediately after leg } j, \forall (j, k) \in A, i \in F(j) \cap F(k) \\ & \text{(note that the nodes } j \text{ and } k \text{ could also correspond to start and end stations,} \\ & \text{respectively, in which case } F(j) \text{ and } F(k) \text{ represent possible } i \in F \\ & \text{that can start and end at such stations, respectively)} \\ 0, & \text{otherwise.} \end{cases}$$

t_j : departure time of flight $j, \forall j \in L$.

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

π_{pqh} : number of passengers in fare-class h on path p who are accepted on path q , $\forall p \in \Pi$, $q \in \Pi_{ph}$, $h \in H_p \cap H_q$ (given that both paths p and q are activated).

Model Formulation:

The *flight scheduling, fleet assignment, and routing model (SFR)* formulated below integrates the schedule design, fleet assignment, and aircraft routing processes, while accommodating additional features such as optional legs, path/itinerary-based demands, flexible flight times, recapture, and multiple fare-classes. This basic model is subsequently tightened in Section 5.2.1 using symmetry-breaking hierarchical constraints, and is also further augmented in Section 5.2.2 with additional restrictions based on aggregate maintenance considerations and in Section 5.2.3 with various classes of valid inequalities.

$$\begin{aligned} \mathbf{SFR} : \text{ Maximize } & \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}} f_{qh} \pi_{pqh} + \sum_{(j,k) \in A_{TF}} \sum_{i \in F(j) \cap F(k)} v_{ijk} y_{ijk} \\ & - \sum_{j \in L} \sum_{i \in F(j)} c_{ij} x_{ij} - \delta \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \pi_{pqh} \end{aligned} \quad (5.2.1)$$

subject to :

$$\sum_{i \in F(j)} x_{ij} = 1, \quad \forall j \in L^M \quad (5.2.2)$$

$$\sum_{i \in F(j)} x_{ij} \leq 1, \quad \forall j \in L^O \quad (5.2.3)$$

$$\sum_{j \in J(i): (n+s, j) \in A} y_{i, n+s, j} \leq 1, \quad \forall i \in F^s, s \in S \quad (5.2.4)$$

$$\begin{aligned} \sum_{i \in F \setminus F^s} \sum_{j: (n+s, j) \in A} y_{i, n+s, j} + \sum_{j: (n+s, j) \in A} \sum_{i \in F^s \setminus F(j)} y_{i, n+s, j} \\ + \sum_{j: (j, n+s) \in A} \sum_{i \in F \setminus F(j)} y_{i, j, n+s} = 0, \quad \forall s \in S \end{aligned} \quad (5.2.5)$$

$$\sum_{i \in F_a^s} \sum_{\substack{j \in J(i): \\ (n+s, j) \in A}} y_{i, n+s, j} = \sum_{i \in F_a} \sum_{\substack{j \in J(i): \\ (j, n+s) \in A}} y_{i, j, n+s}, \quad \forall s \in S, a \in AT \quad (5.2.6)$$

$$\sum_{k: (j, k) \in A, k \in J(i)} y_{ijk} = x_{ij}, \quad \forall j \in L, i \in F(j) \quad (5.2.7)$$

$$\sum_{k: (k, j) \in A, k \in J(i)} y_{ikj} = x_{ij}, \quad \forall j \in L, i \in F(j) \quad (5.2.8)$$

$$x_{ij} - x_{ik} = 0, \quad \forall i \in F(j) \cap F(k), (j, k) \in L^{TF} \quad (5.2.9)$$

$$t_k \geq t_j - b_j + a_k + \sum_{i \in F(j) \cap F(k)} (t_{ij} + \tau_i + b_j - a_k) y_{ijk}, \quad \forall (j, k) \in A, \text{ with } \{j, k\} \subset L \quad (5.2.10)$$

$$a_j \leq t_j \leq b_j, \quad \forall j \in L \quad (5.2.11)$$

$$z_p - \sum_{i \in F(j)} x_{ij} \leq 0, \quad \forall p \in \Pi^O, j \in L^O(p) \quad (5.2.12)$$

$$z_p - \sum_{j \in L^O(p)} \sum_{i \in F(j)} x_{ij} \geq 1 - |L^O(p)|, \quad \forall p \in \Pi^O \quad (5.2.13)$$

$$\sum_{i \in F(j)} x_{ij} \leq \sum_{p \in \Pi^O(j)} z_p, \quad \forall j \in L^O \quad (5.2.14)$$

$$\sum_{p \in \Pi} \sum_{q \in \Pi_{ph} \cap \Pi(j)} \pi_{pqh} \leq \sum_{i \in F(j)} Cap_{ih} x_{ij}, \quad \forall j \in L, h \in H \quad (5.2.15)$$

$$\sum_{q \in \Pi_{ph}} \pi_{pqh} \leq \mu_{ph}, \quad \forall p \in \Pi, h \in H_p \quad (5.2.16)$$

$$\pi_{pph} \leq \mu_{ph} z_p, \quad \forall p \in \Pi^O, h \in H_p \quad (5.2.17)$$

$$\pi_{pqh} \leq \beta_{pqh} (\mu_{ph} - \pi_{pph}) z_q, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph}, \\ q \neq p \quad (z_q \equiv 1, \forall q \in \Pi \setminus \Pi^O) \quad (5.2.18)$$

$$(x, y, z) : \text{ binary}, \quad \pi \geq 0, \quad t : \text{ continuous (see (5.2.11)).} \quad (5.2.19)$$

The objective function (5.2.1) seeks to maximize the net profit as given by the revenue (first two terms) minus the cost (next two terms). The first term represents the direct revenue whereas the second term reflects the additional revenue opportunity for some potential subset

of flight pairs $A_{TF} \subset A$, other than the mandatory through-flight pairs as specified in (5.2.9). The third term accounts for the total aircraft assignment cost, and the last penalty term induces an optimal solution to route passengers on their first path choices whenever possible. Note that, if so desired, we could use an individual penalty term, δ_{pqh} , for each case of rerouting passengers in fare-class h from itinerary p to itinerary q ($q \neq p$). Constraints (5.2.2) and (5.2.3) are the cover constraints for the mandatory legs and the optional legs, respectively. Constraints (5.2.4)-(5.2.8) express that each scheduled flight is included in a single aircraft route that starts at a permissible station, visits a subset of leg-nodes, and ends at a station (which might differ from the starting station.) Constraint (5.2.4) ensures that each aircraft is assigned to at most one route. Constraint (5.2.5) requires that the initial station of an aircraft and the departure station of the first compatible leg on a route must coincide, as must the final station and the arrival station of the last leg. (Equivalently, we can remove these constraints and explicitly restrict all undefined variables $y_{i,n+s,j}$ -variables to zero.) Constraint (5.2.6) enforces that the number of aircraft of any type $a \in AT$ that start at a station $s \in S$ must equal the number of aircraft of this type that wrap around back to this station at the end of the planning horizon (which is a day, assuming the same every-day flight schedule). Note that we tacitly assume here that if all the flights are operated within their respective time-windows, then this feature by itself ensures that there is sufficient ground time overnight to obtain a *periodic schedule* (i.e., one that can be repeated on a daily basis). Constraints (5.2.7) and (5.2.8) serve to define valid routes for each aircraft. Constraint (5.2.9) matches aircraft assignments on certain inbound legs j into a station with that on appropriate outbound legs k at the same station in order to create *through-flights* (i.e., in the present context, flight connections flown by the same aircraft with one stop-over). Airline companies actively seek to schedule through-flights, which generate more revenues because passengers prefer these flights due to their convenience and the assurance of not missing connections. The time-window restrictions are considered in Constraints (5.2.10) and (5.2.11), which can be further lifted as demonstrated in the sequel (see Section 5.2.3). Note that Constraint (5.2.10) is valid since at most one aircraft can be assigned to consecutively serve the legs j and

k , and provides a more compact, yet tighter, representation than the alternative inequalities $t_k \geq t_j - b_j + a_k + (t_{ij} + \tau_i + b_j - a_k)y_{ijk}$, $\forall (j, k) \in A$, with $\{j, k\} \subset L$, $i \in F(j) \cap F(k)$. Constraint (5.2.12) ensures that if an optional leg is excluded from the network, then all paths that contain this leg will also be excluded, and Constraint (5.2.13) 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 (5.2.14) 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. Constraints (5.2.15)-(5.2.18) model the availability and demand constraints, including recapture issues, while considering fare-classes. Constraint (5.2.15) requires that the number of passengers flown on each leg does not exceed the capacity of the aircraft assigned to that leg for each fare-class. Constraint (5.2.16) restricts the number of accepted passengers corresponding to any itinerary/path (including those recaptured on another path) to be no more than the expected demand on that path for each fare-class. Constraint (5.2.17) asserts that the number of passengers served on its own desired path is zero if this path is not activated (being optional), and is less than the original demand for its fare-class if this optional path is activated. Likewise, for those passengers who are not initially accommodated on their original path p , the recapture of this spilled demand on an alternative activated path q (i.e., having $z_q = 1$) is considered in (5.2.18). Note that when an optional path $p \in \Pi^O$ is not activated, all its demand is spilled ($\pi_{pph} = 0$, $\forall h \in H_p$, by (5.2.17)), but some of it might yet be recaptured on other activated paths subject to (5.2.16) and (5.2.18). Also, observe that Constraint (5.2.18) is nonlinear when $q \in \Pi_{ph} \cap \Pi^O$, and will be subsequently linearized in our model reformulation process. Finally, Constraint (5.2.19) imposes logical restrictions on the decision variables.

5.2.1 Symmetry-breaking Constraints

It is important to note that since aircraft of the same type are indistinguishable, the model possesses many equivalent feasible solutions that can be obtained by simply reindexing (or

swapping the resultant routes of) the aircraft of the same type $a \in AT$ that depart from the same initial station $s \in S$, given the common set of legs $L^{a,s}$ that they can potentially serve. Hence, we incorporate suitable hierarchical symmetry-breaking constraints to improve the model solvability. Accordingly, assume that the indices of the aircraft belonging to F_a^s (with $|F_a^s| > 1$) are $\sigma(1, a, s)$, $\sigma(2, a, s)$, ..., $\sigma(|F_a^s|, a, s)$, respectively. Then, we include a hierarchy between the aircraft of the same type by appending to the model the following constraints, where the coefficient j^2 is arbitrarily used to provide a sharper distinction:

$$\sum_{j \in L^{a,s}} j^2 x_{\sigma(m,a,s),j} \geq \sum_{j \in L^{a,s}} j^2 x_{\sigma(m+1,a,s),j}, \quad \forall m = 1, \dots, |F_a^s| - 1, \quad a \in AT, \quad s \in S. \quad (5.2.20)$$

Furthermore, in light of the objective perturbation strategy suggested by Ghoniem and Sherali (2009) to derive *symmetry compatible formulations*, we explore perturbing the objective function (5.2.1) with the term

$$\epsilon_o \tilde{f} \equiv \epsilon_o \sum_{s \in S} \sum_{a \in AT} \sum_{m=1}^{|F_a^s|} \left[w_m \sum_{j \in L^{a,s}} j^2 x_{\sigma(m,a,s),j} \right], \quad (5.2.21)$$

where $w_1 > w_2 > \dots > w_M > 0$ are suitably chosen weights for the different terms in (5.2.20), with $M \equiv \max_{a \in AT, s \in S} |F_a^s|$, and where $\epsilon_o > 0$ is small enough to preserve optimality in Problem SFR with respect to its original objective function. By Sherali and Soyster (1983), this can be ensured (within a unit of optimality) by selecting $\epsilon_o = 1/\tilde{f}_{max}$, where \tilde{f}_{max} is an upper bound on the maximal value of \tilde{f} in (5.2.21), and can be taken as $\tilde{f}_{max} \equiv \sum_{s \in S} \sum_{a \in AT} \sum_{m=1}^{|F_a^s|} [w_m(n - m + 1)^2]$ noting (5.2.2) and (5.2.3) and assuming that $n \geq |F_a^s|$, $\forall s \in S, a \in AT$. Following Ghoniem and Sherali (2009), we utilize $w_m \equiv \frac{1}{am^b + cm}$, $m = 1, \dots, M$, where a , b , and c , are suitable nonnegative scalars.

5.2.2 Aircraft Maintenance Routing Decisions

As mentioned earlier, maintenance requirements are usually addressed at a subsequent stage when solving the aircraft routing problem. However, Sandhu and Klabjan (2007) proposed

the integrated fleet and crew scheduling model and included certain plane-count constraints to obviate potential maintenance-infeasible routings. These constraints were originally introduced by Klabjan et al. (2002) while solving the crew pairing problem to promote feasibility of the subsequently solved aircraft routing problem. The motivation here is that if there exists a pairing that includes a *short-connection* (i.e., a connection that is shorter than the minimum sit-time but long enough to be feasible for a crew if the two sequential flights utilize the same aircraft), then the same aircraft must be used on that connection arc in the aircraft routing network. In order to maintain the feasibility of aircraft routing, the plane-count constraint requires that the number of aircraft (in the routing process) on the ground arc associated with a short-connection should not exceed the corresponding ground arc value from the fleet assignment solution. The authors claimed that maintenance-feasibility can be achieved without difficulty in the case of hub-and-spoke network structures as long as the number of short-connections is low.

In this section, we develop a mechanism to accommodate maintenance requirements in aggregate form within our flight network-based integrated model SFR itself, as opposed to considering maintenance issues only subsequently in an aircraft routing model (see Talluri 1998 and Lan et al. 2006). However, due to the planning nature of considering schedule design and fleet assignment simultaneously with aircraft routing, and noting that we are assuming a daily planning horizon where the flight schedule repeats every day, we simply impose an average daily flight time limit on each type of aircraft, as well as a total flight time limit on the set of aircraft of each given type, in order to facilitate maintenance-feasible routes. To model this consideration within Problem SFR, consider the following additional notation:

$S_M \subseteq S$: stations having maintenance service facilities.

$S_{NM} \equiv S \setminus S_M$: non-maintenance stations.

λ_a : daily (average) flight time limit imposed for aircraft of type $a \in AT$.

λ_a^{tot} : limit imposed on the total flight time for the set of aircraft of type $a \in AT$ ($\lambda_a^{tot} < |F_a| \lambda_a, \forall a \in AT$).

t_{is} : flight time required for aircraft $i \in F$ to fly from station s to a nearest maintenance station ($t_{is} \equiv 0, \forall i \in F, s \in S_M$).

$J^{a,s}(i) = \{j \in J(i) : (n+s, j) \in A \text{ and aircraft } i \text{ can serve flight } j \text{ and then get to a maintenance station if necessary (deadhead in case flight } j \text{ does not lead to a maintenance station) in time not exceeding } \lambda_a\}, \forall i \in F_a^s, a \in AT, s \in S$.

$J_0^{a,s}(i) = \{j \in L : (n+s, j) \in A\} \setminus J^{a,s}(i), \forall i \in F_a^s, a \in AT, s \in S$.

To augment Model SFR, we replace Constraint (5.2.4) by:

$$\sum_{j \in J^{a,s}(i)} y_{i,n+s,j} \leq 1, \quad \forall i \in F_a^s, a \in AT, s \in S, \quad (5.2.22)$$

$$y_{i,n+s,j} = 0, \quad \forall i \in F_a^s, a \in AT, s \in S, j \in J_0^{a,s}(i), \quad (5.2.23)$$

where we limit the potential initial assignment of aircraft $i \in F_a^s$ only to some leg $j \in J^{a,s}(i)$ after serving which it is possible to reach a maintenance station without violating the daily duration limit λ_a . In addition, we impose daily and total flying time limits for each aircraft type to promote maintenance-feasible routings by incorporating the constraints:

$$\sum_{j \in J(i)} t_{ij} x_{ij} + \sum_{s \in S_{NM}} \sum_{\substack{j \in J(i): \\ (j,n+s) \in A}} t_{is} y_{i,j,n+s} \leq \lambda_a, \quad \forall i \in F_a, a \in AT, \quad (5.2.24)$$

$$\sum_{i \in F_a} \left[\sum_{j \in J(i)} t_{ij} x_{ij} + \sum_{s \in S_{NM}} \sum_{\substack{j \in J(i): \\ (j,n+s) \in A}} t_{is} y_{i,j,n+s} \right] \leq \lambda_a^{tot}, \quad \forall a \in AT. \quad (5.2.25)$$

Observe that the symmetry-breaking constraints (5.2.20), which are imposed for aircraft of the same type that depart from the same initial station, still remain valid and appropriate. As far as the prescribed values of λ_a and λ_a^{tot} , $\forall a \in AT$, are concerned, note that Constraints (5.2.24) and (5.2.25) are intended to facilitate maintenance-feasible routings when

subsequently solving the aircraft routing problem. The related FAA regulations require four different types of aircraft maintenance checks (i.e., A, B, C, and D checks) depending on their scale, frequency, and duration. The type A checks are performed most frequently, about every 65 flight hours and require 8 hours of maintenance service, and the type B checks are conducted every 300 to 600 flight hours, whereas the C and D types of checks occur every one to four years for up to a month. Hence, only type A checks need to be considered in our model, since the other checks are conducted less frequently and the aircraft requiring such maintenance checks can be taken out of service and the number of available aircraft of each type can be accordingly adjusted in the model formulation. Considering that A checks are performed every three to four days in practice (based on typical daily aircraft usage), in order to accommodate type A checks within the daily time horizon of our model, we set λ_a equal to 18 (≈ 65 hours / 3.5 days), and use this as a reference value that is further calibrated by multiplying it with a factor f_a that reflects the size of the aircraft type and the average number of years in service. Using this value of $\lambda_a (= 18f_a)$, we let $\lambda_a^{tot} = u_a |F_a| \lambda_a$, where $u_a \in (0, 1)$ is assessed by taking into account historical data of total usage for each aircraft type a . The foregoing parameter values for the maintenance constraints (5.2.24) and (5.2.25) are recommended only as average typical values; whenever more specific maintenance status information of this type is available for the mix of aircraft at the time of implementing the model, the utilized values of λ_a and λ_a^{tot} can be readjusted to promote maintenance-feasible decisions.

5.2.3 Valid Inequalities

We begin our analysis in this section by first lifting the constraints (5.2.10) and (5.2.11), as in Propositions 5.1 and 5.2, respectively, to yield tighter constraints (of the same size) that can be used to replace these original basic restrictions.

Proposition 5.1. The following valid inequalities lift (5.2.11) and can therefore replace

these restrictions:

$$t_j \geq a_j + \sum_{i \in F(j)} \sum_{\substack{k: (k,j) \in A \\ k \in J(i)}} [\max\{0, a_k + t_{ik} + \tau_i - a_j\} y_{ikj}], \quad \forall j \in L \quad (5.2.26)$$

$$t_j \leq b_j - \sum_{i \in F(j)} \sum_{\substack{k: (j,k) \in A \\ k \in J(i)}} [\max\{0, b_j + t_{ij} + \tau_i - b_k\} y_{ijk}], \quad \forall j \in L. \quad (5.2.27)$$

Proof: Since the lifting is evident, it is sufficient to establish the validity of (5.2.26) and (5.2.27). First, consider (5.2.26) for any given $j \in L$. Observe from (5.2.2), (5.2.3), and (5.2.8) that we have

$$\Delta_1 \equiv \sum_{i \in F(j)} \sum_{\substack{k: (k,j) \in A \\ k \in J(i)}} y_{ikj} = \sum_{i \in F(j)} x_{ij} \leq 1. \quad (5.2.28)$$

Hence, if $\Delta_1 = 0$ in (5.2.28), then (5.2.26) is trivially valid; else, we have $\Delta_1 = 1$ with $y_{ikj} = 1$ for some $i \in F(j)$ and $k \in J(i)$ such that $(k, j) \in A$, with the remaining y -variables in (5.2.28) being zero. In this case, (5.2.26) asserts that

$$t_j \geq a_j + \max\{0, a_k + t_{ik} + \tau_i - a_j\} = \max\{a_j, a_k + t_{ik} + \tau_i\}. \quad (5.2.29)$$

But when $y_{ikj} = 1$, we know from (5.2.10) and (5.2.11) that $t_j \geq a_j$ and $t_j \geq t_k + t_{ik} + \tau_i \geq a_k + t_{ik} + \tau_i$, i.e., $t_j \geq \max\{a_j, a_k + t_{ik} + \tau_i\}$, which establishes the validity of (5.2.29), and therefore that of (5.2.26).

Next, consider (5.2.27) for any given $j \in L$. Similar to above, we have from (5.2.2), (5.2.3), and (5.2.7) that

$$\Delta_2 \equiv \sum_{i \in F(j)} \sum_{\substack{k: (j,k) \in A \\ k \in J(i)}} y_{ijk} = \sum_{i \in F(j)} x_{ij} \leq 1. \quad (5.2.30)$$

Hence, if $\Delta_2 = 0$ in (5.2.30), then (5.2.27) is trivially valid; else, we have $\Delta_2 = 1$ with $y_{ijk} = 1$ for some $i \in F(j)$ and $k \in J(i)$ such that $(j, k) \in A$, with the remaining y -variables in (5.2.30) being zero. In this case, (5.2.27) asserts that

$$\begin{aligned} t_j &\leq b_j - \max\{0, b_j + t_{ij} + \tau_i - b_k\} = b_j + \min\{0, b_k - b_j - t_{ij} - \tau_i\} \\ &= \min\{b_j, b_k - t_{ij} - \tau_i\}. \end{aligned} \quad (5.2.31)$$

But when $y_{ijk} = 1$, we know from (5.2.10) and (5.2.11) that $t_j \leq b_j$ and $b_k \geq t_k \geq t_j + t_{ij} + \tau_i$, i.e., $t_j \leq b_k - t_{ij} - \tau_i$, and so, $t_j \leq \min\{b_j, b_k - t_{ij} - \tau_i\}$, which establishes the validity of (5.2.31), and therefore that of (5.2.27). \square

Proposition 5.2. The following valid inequalities lift (5.2.10) and can therefore replace these restrictions:

$$\begin{aligned}
 t_k \geq & t_j - b_j + a_k + \sum_{i \in F(j) \cap F(k)} [\max\{0, t_{ij} + \tau_i + b_j - a_k\} y_{ijk}] \\
 & + \sum_{i \in F(k)} \sum_{\substack{h: (h,k) \in A \\ h \in J(i), h \neq j}} [\max\{0, a_h + t_{ih} + \tau_i - a_k\} y_{ihk}] \\
 & + \sum_{i \in F(j)} \sum_{\substack{l: (j,l) \in A \\ l \in J(i), l \neq k}} [\max\{0, b_j + t_{ij} + \tau_i - b_l\} y_{ijl}], \\
 & \forall (j, k) \in A, \text{ with } \{j, k\} \subset L. \quad (5.2.32)
 \end{aligned}$$

Proof: Consider any $(j, k) \in A$, with $\{j, k\} \subset L$. Suppose that $y_{ijk} = 1$ for some $i \in F(j) \cap F(k)$. Then $y_{ihk} = 0$, $\forall h : (h, k) \in A, h \neq j$ by (5.2.2), (5.2.3), and (5.2.8), and $y_{ijl} = 0$, $\forall l : (j, l) \in A, l \neq k$ by (5.2.2), (5.2.3), and (5.2.7). Therefore, (5.2.32) asserts that

$$\begin{aligned}
 t_k & \geq t_j - b_j + a_k + \max\{0, t_{ij} + \tau_i + b_j - a_k\} \\
 & = \max\{t_j - b_j + a_k, t_j + t_{ij} + \tau_i\}, \quad (5.2.33)
 \end{aligned}$$

which is valid because from (5.2.11) and (5.2.10), we respectively have that $t_k \geq t_j - b_j + a_k$ and $t_k \geq t_j + t_{ij} + \tau_i$, i.e., $t_k \geq \max\{t_j - b_j + a_k, t_j + t_{ij} + \tau_i\}$.

On the other hand, if $y_{ijk} = 0$, $\forall i \in F(j) \cap F(k)$, then (5.2.32) asserts that

$$\begin{aligned}
 t_k - t_j \geq & a_k - b_j + \sum_{i \in F(k)} \sum_{\substack{h: (h,k) \in A \\ h \in J(i), h \neq j}} [\max\{0, a_h + t_{ih} + \tau_i - a_k\} y_{ihk}] \\
 & + \sum_{i \in F(j)} \sum_{\substack{l: (j,l) \in A \\ l \in J(i), l \neq k}} [\max\{0, b_j + t_{ij} + \tau_i - b_l\} y_{ijl}]. \quad (5.2.34)
 \end{aligned}$$

In this case, we have from (5.2.26) that,

$$\begin{aligned}
t_k &\geq a_k + \sum_{i \in F(k)} \sum_{\substack{h: (h,k) \in A \\ h \in J(i)}} [\max\{0, a_h + t_{ih} + \tau_i - a_k\} y_{ihk}] \\
&= a_k + \sum_{i \in F(k)} \sum_{\substack{h: (h,k) \in A \\ h \in J(i), h \neq j}} [\max\{0, a_h + t_{ih} + \tau_i - a_k\} y_{ihk}] \tag{5.2.35}
\end{aligned}$$

and we have from (5.2.27) that,

$$\begin{aligned}
-t_j &\geq -b_j + \sum_{i \in F(j)} \sum_{\substack{l: (j,l) \in A \\ l \in J(i)}} [\max\{0, b_j + t_{ij} + \tau_i - b_l\} y_{ijl}] \\
&= -b_j + \sum_{i \in F(j)} \sum_{\substack{l: (j,l) \in A \\ l \in J(i), l \neq k}} [\max\{0, b_j + t_{ij} + \tau_i - b_l\} y_{ijl}]. \tag{5.2.36}
\end{aligned}$$

Summing (5.2.35) and (5.2.36) yields (5.2.34), which therefore establishes the validity of (5.2.32). \square

Next, we identify various classes of valid inequalities to include in Model SFR, as motivated by the development in Chapters 3 and 4. Proposition 5.3 below is based on the Reformulation-Linearization Technique (RLT) (Sherali and Adams, 1999) and provides a linearization of (5.2.18) for the nonlinear case when $q \in \Pi_{ph} \cap \Pi^O$, where π_{pphq} is a new RLT-variable that represents the product term $\pi_{pph} z_q$, $\forall p \in \Pi$, $h \in H_p$, $q \in \Pi_{ph} \cap \Pi^O$, $q \neq p$.

Proposition 5.3. The following valid linear inequalities can be used to equivalently replace Constraint (5.2.18):

$$\pi_{pqh} \leq \beta_{pqh} \mu_{ph} - \beta_{pqh} \pi_{pph}, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph} \setminus \Pi^O, q \neq p, \tag{5.2.37a}$$

$$\pi_{pqh} \leq \beta_{pqh} \mu_{ph} z_q - \beta_{pqh} \pi_{pphq}, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph} \cap \Pi^O, q \neq p, \tag{5.2.37b}$$

where we additionally have

$$\pi_{pphq} \geq \pi_{pph} - \mu_{ph}(1 - z_q) \text{ and } \pi_{pphq} \geq 0, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph} \cap \Pi^O, q \neq p. \tag{5.2.37c}$$

Propositions 5.4-5.5 present additional valid inequalities to tighten the model representation based on certain constraint structures related to demand and capacity.

Proposition 5.4. The following is a valid inequality:

$$\sum_{p \in \Pi(j)} \pi_{pph} \leq \sum_{i \in F(j)} \widetilde{Cap}_{ihj} x_{ij}, \quad \forall j \in L, h \in H, \quad (5.2.38)$$

where

$$\widetilde{Cap}_{ihj} \equiv \min\{Cap_{ih}, \sum_{p \in \Pi(j)} \mu_{ph}\}, \quad \forall j \in L, i \in F(j), h \in H.$$

Proposition 5.5. The following is a valid inequality that can replace (5.2.17):

$$\pi_{pph} \leq \tilde{\mu}_{ph} z_p, \quad \forall p \in \Pi^O, h \in H_p, \quad (5.2.39)$$

where

$$\tilde{\mu}_{ph} \equiv \min\{\mu_{ph}, \max_{i \in \cup_{j \in L(p)} F(j)} Cap_{ih}\}, \quad \forall p \in \Pi^O, h \in H_p.$$

Finally, Proposition 5.6 presents another class of valid inequalities that we adopt from Chapter 3, along with its associated separation routine as described below, to further tighten the model representation.

Proposition 5.6. For any $j^* \in L^{O+} \equiv \{j \in L^O : |L^O(p)| \geq 2, \forall p \in \Pi^O(j)\}$, let $S_p^{j^*} \equiv L^O(p) \setminus \{j^*\}$, $\forall p \in \Pi^O(j^*)$, and compute $\nu^{j^*} = \min_{p \in \Pi^O(j^*)} |S_p^{j^*}|$. Select $S^{j^*} \subseteq S_{\cup}^{j^*} \equiv \cup_{p \in \Pi^O(j^*)} S_p^{j^*}$ such that $|S^{j^*} \cap S_p^{j^*}| \geq \nu^{j^*}, \forall p \in \Pi^O(j^*)$. Then the following is a valid inequality:

$$\sum_{j \in S^{j^*}} \sum_{i \in F(j)} x_{ij} \geq \nu^{j^*} \sum_{i \in F(j^*)} x_{ij^*}. \quad (5.2.40)$$

As described in Chapter 3, the following *separation problem* can be used to possibly generate a valid inequality of the type (5.2.40) that deletes the current LP relaxation solution

$(\bar{x}, \bar{y}, \bar{z}, \bar{\pi}, \bar{t})$ to Problem SFR, where the objective function (5.2.41a) determines the left-hand side of (5.2.40) at the given LP solution, and Constraint (5.2.41b) represents the restrictions on S^{j^*} as per Proposition 5.6, and where the binary variable ζ_j takes on a value of 1 if $j \in S^{j^*}$ and 0 otherwise, $\forall j \in S_{\cup}^{j^*}$:

$$\mathbf{SEP}(j^*) : \text{Minimize} \quad \sum_{j \in S_{\cup}^{j^*}} \left[\sum_{i \in F(j)} \bar{x}_{ij} \right] \zeta_j \quad (5.2.41a)$$

$$\text{subject to:} \quad \sum_{j \in S_p^{j^*}} \zeta_j \geq \nu^{j^*}, \quad \forall p \in \Pi^O(j^*) \quad (5.2.41b)$$

$$\zeta_j : \text{binary}, \quad \forall j \in S_{\cup}^{j^*}. \quad (5.2.41c)$$

If the optimal objective value for Problem $\mathbf{SEP}(j^*)$ is less than $\nu^{j^*} \sum_{i \in F(j^*)} \bar{x}_{ij^*}$ for the selected $j^* \in L^{O+}$, then the cut (5.2.40) generated via the corresponding ζ -solution will delete the current LP relaxation solution to Model SFR.

5.3 Algorithmic Solution Approach

In this section, we describe three algorithmic solution procedures for solving Model SFR. In the first of these discussed below (**Algorithm A1**), we linearize the model and accommodate maintenance-feasibility constraints along with the proposed sets of valid inequalities to derive a lifted linear mixed-integer programming representation. Next, we apply Benders decomposition to a suitable relaxation of the resulting model to fix leg selection and flight timing decisions, and finally, we recover fleet assignment, aircraft routing, and demand allocation decisions through a reduced resultant model. This process therefore enables an integrated look-ahead viewpoint in making the strategic leg selection and flight scheduling decisions, while retaining computational tractability.

Algorithm A1:

(Step A1.1) Linearization, Maintenance Consideration, and Model Enhancements: We first linearize Model SFR by replacing Constraint (5.2.18) with the inequalities (5.2.37a)-(5.2.37c) from Proposition 5.3. Next, we incorporate the aforementioned symmetry-breaking constraints (5.2.20) and the corresponding objective perturbation strategy (see Equation (5.2.21)) from Section 5.2.1 within the model, along with the maintenance-feasible routing constraints (5.2.22)-(5.2.25) as in Section 5.2.2, the lifted versions (5.2.26), (5.2.27), and (5.2.32) of Constraints (5.2.11) and (5.2.10) from Propositions 5.1 and 5.2, respectively, and the valid inequalities (5.2.38) and (5.2.39) from Propositions 5.4 and 5.5, respectively. Finally, we solve the LP relaxation of the resulting model and sequentially generate rounds of cuts (5.2.40) given by Proposition 5.6 via the separation problem (5.2.41), as discussed in Section 5.2.3. Now, define $\xi_j \equiv \sum_{i \in F(j)} x_{ij}$, $\forall j \in L^O$ and denote the cuts (5.2.40) generated via the separation routine as $A\xi \leq b$, where $\xi \equiv (\xi_j, j \in L^O)$. The enhanced Model SFR can then be re-written as follows:

$$\begin{aligned} \text{SFR}^+ : \quad & \text{Maximize} \quad \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}} f_{qh} \pi_{pqh} + \sum_{(j,k) \in AT_F} \sum_{i \in F(j) \cap F(k)} v_{ijk} y_{ijk} - \sum_{j \in L} \sum_{i \in F(j)} c_{ij} x_{ij} \\ & - \delta \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \pi_{pqh} + \epsilon_o \sum_{s \in S} \sum_{a \in AT} \sum_{m=1}^{|F_a^s|} \left[w_m \sum_{j \in L^{a,s}} j^2 x_{\sigma(m,a,s),j} \right] \end{aligned} \quad (5.3.1)$$

subject to:

$$\sum_{i \in F(j)} x_{ij} = 1, \quad \forall j \in L^M \quad (5.3.2)$$

$$\xi_j \leq 1, \quad \forall j \in L^O \quad (5.3.3)$$

$$\sum_{j \in J^{a,s}(i)} y_{i,n+s,j} \leq 1, \quad \forall i \in F_a^s, a \in AT, s \in S \quad (5.3.4)$$

$$y_{i,n+s,j} = 0, \quad \forall i \in F_a^s, a \in AT, s \in S, j \in J_0^{a,s}(i) \quad (5.3.5)$$

$$\sum_{i \in F \setminus F^s} \sum_{j: (n+s,j) \in A} y_{i,n+s,j} + \sum_{j: (n+s,j) \in A} \sum_{i \in F^s \setminus F(j)} y_{i,n+s,j}$$

$$+ \sum_{j:(j,n+s) \in A} \sum_{i \in F \setminus F(j)} y_{i,j,n+s} = 0, \quad \forall s \in S \quad (5.3.6)$$

$$\sum_{i \in F_a^s} \sum_{\substack{j \in J(i): \\ (n+s,j) \in A}} y_{i,n+s,j} = \sum_{i \in F_a} \sum_{\substack{j \in J(i): \\ (j,n+s) \in A}} y_{i,j,n+s}, \quad \forall s \in S, a \in AT \quad (5.3.7)$$

$$\sum_{k:(j,k) \in A, k \in J(i)} y_{ijk} = x_{ij}, \quad \forall j \in L, i \in F(j) \quad (5.3.8)$$

$$\sum_{k:(k,j) \in A, k \in J(i)} y_{ikj} = x_{ij}, \quad \forall j \in L, i \in F(j) \quad (5.3.9)$$

$$x_{ij} - x_{ik} = 0, \quad \forall i \in F(j) \cap F(k), (j,k) \in L^{TF} \quad (5.3.10)$$

$$\begin{aligned} t_k &\geq t_j - b_j + a_k + \sum_{i \in F(j) \cap F(k)} [\max\{0, t_{ij} + \tau_i + b_j - a_k\} y_{ijk}] \\ &\quad + \sum_{i \in F(k)} \sum_{\substack{h:(h,k) \in A \\ h \in J(i), h \neq j}} [\max\{0, a_h + t_{ih} + \tau_i - a_k\} y_{ihk}] \\ &+ \sum_{i \in F(j)} \sum_{\substack{l:(j,l) \in A \\ l \in J(i), l \neq k}} [\max\{0, b_j + t_{ij} + \tau_i - b_l\} y_{ijl}], \quad \forall (j,k) \in A, \text{ with } \{j,k\} \subset L \end{aligned} \quad (5.3.11)$$

$$t_j \geq a_j + \sum_{i \in F(j)} \sum_{\substack{k:(k,j) \in A \\ k \in J(i)}} [\max\{0, a_k + t_{ik} + \tau_i - a_j\} y_{ikj}], \quad \forall j \in L \quad (5.3.12)$$

$$t_j \leq b_j - \sum_{i \in F(j)} \sum_{\substack{k:(j,k) \in A \\ k \in J(i)}} [\max\{0, b_j + t_{ij} + \tau_i - b_k\} y_{ijk}], \quad \forall j \in L \quad (5.3.13)$$

$$z_p \leq \xi_j, \quad \forall p \in \Pi^O, j \in L^O(p) \quad (5.3.14)$$

$$z_p - \sum_{j \in L^O(p)} \xi_j \geq 1 - |L^O(p)|, \quad \forall p \in \Pi^O \quad (5.3.15)$$

$$\xi_j \leq \sum_{p \in \Pi^O(j)} z_p, \quad \forall j \in L^O \quad (5.3.16)$$

$$\sum_{p \in \Pi} \sum_{q \in \Pi_{ph} \cap \Pi(j)} \pi_{pqh} \leq \sum_{i \in F(j)} Cap_{ih} x_{ij}, \quad \forall j \in L, h \in H \quad (5.3.17)$$

$$\sum_{p \in \Pi(j)} \pi_{pph} \leq \sum_{i \in F(j)} \widetilde{Cap}_{ihj} x_{ij}, \quad \forall j \in L, h \in H \quad (5.3.18)$$

$$\sum_{q \in \Pi_{ph}} \pi_{pqh} \leq \mu_{ph}, \quad \forall p \in \Pi, h \in H_p \quad (5.3.19)$$

$$\pi_{pph} \leq \tilde{\mu}_{ph} z_p, \quad \forall p \in \Pi^O, h \in H_p \quad (5.3.20)$$

$$\pi_{pqh} + \beta_{pqh} \pi_{pph} \leq \beta_{pqh} \mu_{ph}, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph} \setminus \Pi^O, q \neq p \quad (5.3.21)$$

$$\pi_{pqh} + \beta_{pqh}\pi_{pphq} \leq \beta_{pqh}\mu_{ph}z_q, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph} \cap \Pi^O, q \neq p \quad (5.3.22)$$

$$\pi_{pph} - \pi_{pphq} \leq \mu_{ph}(1 - z_q), \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph} \cap \Pi^O, q \neq p \quad (5.3.23)$$

$$\sum_{j \in L^{a,s}} j^2 x_{\sigma(m,a,s),j} \geq \sum_{j \in L^{a,s}} j^2 x_{\sigma(m+1,a,s),j}, \quad \forall m = 1, \dots, |F_a^s| - 1, a \in AT, s \in S \quad (5.3.24)$$

$$\sum_{j \in J(i)} t_{ij}x_{ij} + \sum_{s \in S_{NM}} \sum_{\substack{j \in J(i): \\ (j,n+s) \in A}} t_{is}y_{i,j,n+s} \leq \lambda_a, \quad \forall i \in F_a, a \in AT \quad (5.3.25)$$

$$\sum_{i \in F_a} \left[\sum_{j \in J(i)} t_{ij}x_{ij} + \sum_{s \in S_{NM}} \sum_{\substack{j \in J(i): \\ (j,n+s) \in A}} t_{is}y_{i,j,n+s} \right] \leq \lambda_a^{tot}, \quad \forall a \in AT \quad (5.3.26)$$

$$A\xi \leq b, \quad (5.3.27)$$

$$\xi_j = \sum_{i \in F(j)} x_{ij}, \quad \forall j \in L^O \quad (5.3.28)$$

$$(x, y, z) : \text{binary}, \quad \pi \geq 0, \quad t : \text{continuous}, \quad \xi_j \geq 0, \quad \forall j \in L^O. \quad (5.3.29)$$

(Step A1.2) Applying Benders Decomposition: We now relax the binary restrictions on the (x, y) -variables in Model SFR^+ and denote the resulting model as $\overline{\text{SFR}^+}$. The binary restrictions on the z -variables imply that ξ will also be binary-valued for any feasible solution to $\overline{\text{SFR}^+}$ by (5.3.14), (5.3.16), and (5.3.29), as proven in Chapter 3. We then apply Benders decomposition to $\overline{\text{SFR}^+}$ by constructing a master program that projects the latter problem onto the space of the (z, ξ) -variables, where a new variable η_0 represents the value function of the remainder of the problem for any fixed (z, ξ) , as determined via the subproblem formulated below. This yields the following decomposition:

(Master Program)

$$\text{MP: Maximize } \eta_0$$

subject to:

$$\eta_0 \leq BC(\lambda, \alpha, u, \sigma, \omega, \rho, \gamma, \tau, \theta, \psi, \varphi),$$

$$\forall (\lambda, \alpha, u, \sigma, \omega, \rho, \gamma, \tau, \theta, \psi, \varphi) \in \nabla \quad (5.3.30)$$

$$\xi_j \leq 1, \quad \forall j \in L^O \quad (5.3.31)$$

$$z_p \leq \xi_j, \quad \forall p \in \Pi^O, \forall j \in L^O(p) \quad (5.3.32)$$

$$z_p - \sum_{j \in L^O(p)} \xi_j \geq 1 - |L^O(p)|, \quad \forall p \in \Pi^O \quad (5.3.33)$$

$$\xi_j \leq \sum_{p \in \Pi^O(j)} z_p, \quad \forall j \in L^O \quad (5.3.34)$$

$$A\xi \leq b, \quad (5.3.35)$$

$$z : \text{binary}, \quad \xi_j \geq 0, \forall j \in L^O, \quad \eta_0 : \text{unrestricted}, \quad (5.3.36)$$

where the Benders cuts (5.3.30) are derived from the dual to the subproblem defined below, and where ∇ represents the set of extreme points of the polyhedron defined by the constraints of this dual subproblem. In order to assure feasibility (and hence primal-dual optimality) in the subproblem given below, we use an artificial variable vector $x^a \equiv (x_j^a, j \in L)$ within the set of constraints (5.3.38) and (5.3.39) along with an associated sufficiently large negative coefficient $-M$ in the objective function (5.3.37).

(Subproblem)

$$\begin{aligned} \text{PSP}(\bar{z}, \bar{\xi}) : \text{Maximize } & \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}} f_{qh} \pi_{pqh} + \sum_{(j,k) \in ATF} \sum_{i \in F(j) \cap F(k)} v_{ijk} y_{ijk} - \sum_{j \in L} \sum_{i \in F(j)} c_{ij} x_{ij} \\ & - \delta \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \pi_{pqh} + \epsilon_o \sum_{s \in S} \sum_{a \in AT} \sum_{m=1}^{|F_a^s|} \left[w_m \sum_{j \in L^{a,s}} j^2 x_{\sigma(m,a,s),j} \right] - M e^T x^a \end{aligned} \quad (5.3.37)$$

subject to:

$$\sum_{i \in F(j)} x_{ij} + x_j^a = 1, \quad \forall j \in L^M \quad (\lambda_j) \quad (5.3.38)$$

$$\sum_{i \in F(j)} x_{ij} + x_j^a = \bar{\xi}_j, \quad \forall j \in L^O \quad (\lambda_j) \quad (5.3.39)$$

$$\sum_{j \in J^{a,s}(i)} y_{i,n+s,j} \leq 1, \quad \forall i \in F_a^s, a \in AT, s \in S \quad (\alpha_{ias}^1) \quad (5.3.40)$$

$$y_{i,n+s,j} = 0, \quad \forall i \in F_a^s, a \in AT, s \in S, j \in J_0^{a,s}(i) \quad (\alpha_{iasj}^2) \quad (5.3.41)$$

$$\begin{aligned} \sum_{i \in F \setminus F^s} \sum_{j: (n+s, j) \in A} y_{i, n+s, j} + \sum_{j: (n+s, j) \in A} \sum_{i \in F^s \setminus F(j)} y_{i, n+s, j} \\ + \sum_{j: (j, n+s) \in A} \sum_{i \in F \setminus F(j)} y_{i, j, n+s} = 0, \quad \forall s \in S \quad (u_s^1) \end{aligned} \quad (5.3.42)$$

$$\sum_{i \in F_a^s} \sum_{\substack{j \in J(i): \\ (n+s, j) \in A}} y_{i, n+s, j} - \sum_{i \in F_a} \sum_{\substack{j \in J(i): \\ (j, n+s) \in A}} y_{i, j, n+s} = 0, \quad \forall s \in S, a \in AT \quad (u_{a_s}^2) \quad (5.3.43)$$

$$\sum_{k: (j, k) \in A, k \in J(i)} y_{ijk} - x_{ij} = 0, \quad \forall j \in L, i \in F(j) \quad (\sigma_{ij}^1) \quad (5.3.44)$$

$$\sum_{k: (k, j) \in A, k \in J(i)} y_{ikj} - x_{ij} = 0, \quad \forall j \in L, i \in F(j) \quad (\sigma_{ij}^2) \quad (5.3.45)$$

$$x_{ij} - x_{ik} = 0, \quad \forall i \in F(j) \cap F(k), (j, k) \in L^{TF} \quad (\omega_{ijk}^1) \quad (5.3.46)$$

$$\begin{aligned} t_j - t_k + \sum_{i \in F(j) \cap F(k)} [\max\{0, t_{ij} + \tau_i + b_j - a_k\} y_{ijk}] \\ + \sum_{i \in F(k)} \sum_{\substack{h: (h, k) \in A \\ h \in J(i), h \neq j}} [\max\{0, a_h + t_{ih} + \tau_i - a_k\} y_{ihk}] \\ + \sum_{i \in F(j)} \sum_{\substack{l: (j, l) \in A \\ l \in J(i), l \neq k}} [\max\{0, b_j + t_{ij} + \tau_i - b_l\} y_{ijl}] \leq b_j - a_k, \\ \forall (j, k) \in A, \text{ with } \{j, k\} \subset L \quad (\omega_{jk}^2) \end{aligned} \quad (5.3.47)$$

$$\sum_{i \in F(j)} \sum_{\substack{k: (k, j) \in A \\ k \in J(i)}} [\max\{0, a_k + t_{ik} + \tau_i - a_j\} y_{ikj}] - t_j \leq -a_j, \quad \forall j \in L \quad (\rho_j^1) \quad (5.3.48)$$

$$\sum_{i \in F(j)} \sum_{\substack{k: (j, k) \in A \\ k \in J(i)}} [\max\{0, b_j + t_{ij} + \tau_i - b_k\} y_{ijk}] + t_j \leq b_j, \quad \forall j \in L \quad (\rho_j^2) \quad (5.3.49)$$

$$\sum_{p \in \Pi} \sum_{q \in \Pi_{ph} \cap \Pi(j)} \pi_{pqh} - \sum_{i \in F(j)} Cap_{ih} x_{ij} \leq 0, \quad \forall j \in L, \forall h \in H \quad (\gamma_{jh}^1) \quad (5.3.50)$$

$$\sum_{p \in \Pi(j)} \pi_{pph} - \sum_{i \in F(j)} \widetilde{Cap}_{ih} x_{ij} \leq 0, \quad \forall j \in L, \forall h \in H \quad (\gamma_{jh}^2) \quad (5.3.51)$$

$$\sum_{q \in \Pi_{ph}} \pi_{pqh} \leq \mu_{ph}, \quad \forall p \in \Pi, \forall h \in H_p \quad (\tau_{ph}^1) \quad (5.3.52)$$

$$\pi_{pph} \leq \tilde{\mu}_{ph} \bar{z}_p, \quad \forall p \in \Pi^O, \forall h \in H_p \quad (\tau_{ph}^2) \quad (5.3.53)$$

$$\pi_{pqh} + \beta_{pqh} \pi_{pph} \leq \beta_{pqh} \mu_{ph}, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph} \setminus \Pi^O, q \neq p \quad (\theta_{pqh}^1) \quad (5.3.54)$$

$$\pi_{pqh} + \beta_{pqh} \pi_{pph} \leq \beta_{pqh} \mu_{ph} \bar{z}_q, \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph} \cap \Pi^O, q \neq p \quad (\theta_{pqh}^1) \quad (5.3.55)$$

$$\pi_{pph} - \pi_{pphq} \leq \mu_{ph}(1 - \bar{z}_q), \quad \forall p \in \Pi, h \in H_p, q \in \Pi_{ph} \cap \Pi^O, q \neq p \quad (\theta_{pqh}^2) \quad (5.3.56)$$

$$\sum_{j \in L^{a,s}} j^2 x_{\sigma(m+1,a,s),j} - \sum_{j \in L^{a,s}} j^2 x_{\sigma(m,a,s),j} \leq 0, \quad \forall m = 1, \dots, |F_a^s| - 1, \\ a \in AT, s \in S \quad (\psi_{mas}) \quad (5.3.57)$$

$$\sum_{j \in J(i)} t_{ij} x_{ij} + \sum_{s \in S_{NM}} \sum_{\substack{j \in J(i): \\ (j,n+s) \in A}} t_{is} y_{i,j,n+s} \leq \lambda_a, \quad \forall i \in F_a, a \in AT \quad (\varphi_{ia}^1) \quad (5.3.58)$$

$$\sum_{i \in F_a} \sum_{j \in J(i)} t_{ij} x_{ij} + \sum_{i \in F_a} \sum_{s \in S_{NM}} \sum_{\substack{j \in J(i): \\ (j,n+s) \in A}} t_{is} y_{i,j,n+s} \leq \lambda_a^{tot}, \quad \forall a \in AT \quad (\varphi_a^2) \quad (5.3.59)$$

$$(x, y, \pi, x^a) \geq 0, \quad t : \text{continuous}, \quad (5.3.60)$$

where the corresponding dual variables are displayed above next to each associated subproblem constraint. Note that $(x, y, \pi) = (0, 0, 0)$; $x_j^a = 1, \forall j \in L^M$; $x_j^a = \bar{\xi}_j, \forall j \in L^O$; and $t_j \in [a_j, b_j], \forall j \in L$, is a feasible solution to Problem $\text{PSP}(\bar{z}, \bar{\xi})$, and so by the boundedness of the (x, y, π, x^a, t) -variables in this problem, an optimum exists. Having obtained an optimum $(\bar{\eta}_0, \bar{z}, \bar{\xi})$ to the master program MP, and having solved the subproblem $\text{PSP}(\bar{z}, \bar{\xi})$ and derived a complementary dual optimal solution $(\bar{\lambda}, \bar{\alpha}, \bar{u}, \bar{\sigma}, \bar{\omega}, \bar{\rho}, \bar{\gamma}, \bar{\tau}, \bar{\theta}, \bar{\psi}, \bar{\varphi})$, in case $\bar{\eta}_0 \leq (1 + \epsilon) \nu[\text{PSP}(\bar{z}, \bar{\xi})]$ for some optimality tolerance $\epsilon \geq 0$, where $\nu[\text{P}]$ denotes the optimal objective value of any problem P, we terminate the algorithmic process. Otherwise, we generate a Benders cut (5.3.30) as follows to delete $(\bar{z}, \bar{\xi})$:

$$\begin{aligned} \eta_0 \leq & \sum_{j \in L^O} \bar{\lambda}_j \bar{\xi}_j + \sum_{p \in \Pi^O} \left[\sum_{h \in H_p} \tilde{\mu}_{ph} \bar{\tau}_{ph}^2 \right] z_p + \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{\substack{q \in \Pi_{ph} \cap \Pi^O \\ :q \neq p}} [\mu_{ph} \beta_{pqh} \bar{\theta}_{pqh}^1 - \mu_{ph} \bar{\theta}_{pqh}^2] z_q \\ & + \left[\sum_{j \in L^M} \bar{\lambda}_j + \sum_{s \in S} \sum_{a \in AT} \sum_{i \in F_a^s} \bar{\alpha}_{ias}^1 + \sum_{\substack{(j,k) \in A \\ : \{j,k\} \subset L}} (b_j - a_k) \bar{\omega}_{jk}^2 + \sum_{j \in L} (b_j \bar{\rho}_j^2 - a_j \bar{\rho}_j^1) + \sum_{p \in \Pi} \sum_{h \in H_p} \mu_{ph} \bar{\tau}_{ph}^1 \right. \\ & \left. + \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{\substack{q \in \Pi_{ph} \setminus \Pi^O \\ :q \neq p}} \beta_{pqh} \mu_{ph} \bar{\theta}_{pqh}^1 + \sum_{p \in \Pi} \sum_{h \in H_p} \sum_{\substack{q \in \Pi_{ph} \cap \Pi^O \\ :q \neq p}} \mu_{ph} \bar{\theta}_{pqh}^2 + \sum_{a \in AT} \sum_{i \in F_a} \lambda_a \bar{\varphi}_{ia}^1 + \sum_{a \in AT} \lambda_a^{tot} \bar{\varphi}_a^2 \right]. \end{aligned} \quad (5.3.61)$$

Remark 5.2. In order to generate strong or nondominated Benders cuts as recommended by Magnanti and Wong (1981), we differentiate among alternative optimal dual solutions to Problem $\text{PSP}(\bar{z}, \bar{\xi})$ by using the following perturbation strategy proposed by Sherali and Lunday (2010), which has the advantage of solving a single subproblem having the same structure as above. Let $(\hat{z}, \hat{\xi}) = e$, where e is an appropriately dimensioned vector of ones, and let \hat{b} denote the right-hand side of the subproblem $\text{PSP}(\bar{z}, \bar{\xi})$ when $(\bar{z}, \bar{\xi})$ is replaced by $(\hat{z}, \hat{\xi}) \equiv e$. Then, while solving $\text{PSP}(\bar{z}, \bar{\xi})$, we perturb its right-hand side (and so the objective function of the dual subproblem) by adding to it $\epsilon_P \hat{b}$, where $\epsilon_P > 0$ is sufficiently small such that the perturbation preserves (near-) optimality with respect to the original dual subproblem. In theory, Sherali and Lunday (2010) derive such a value of ϵ_P and show that the resulting Benders cut is valid and nondominated with respect to a set of alternative (near-)optimal dual solutions. However, since the magnitude of such a theoretical value can be much too small for a practical implementation, we utilize $\epsilon_P \equiv 10^{-6}$ in our computations. Nonetheless, note that since the perturbed subproblem generates a dual feasible solution with respect to the original subproblem (if not a guaranteed dual optimal solution), the generated Benders cut remains valid under such a perturbation. \square

(Step A1.3) Formulation of Reduced Problem (FRR): Having thus obtained a (partial) solution (z^*, ξ^*) to $\overline{\text{SFR}^+}$, we now fix $(z, \xi) \equiv (z^*, \xi^*)$ and thereby ascertain the optional legs and paths (itineraries) to activate, deleting the remaining ostensibly unprofitable legs. Let $\hat{\Pi} \subseteq \Pi$ and $\hat{L} \subseteq L$ respectively denote the set of paths and the set of legs resulting from this fixing procedure, and let $\hat{L}^{TF}, \hat{A}_{TF}, \hat{\Pi}_{ph}, \hat{A}, \hat{J}(i), \hat{L}^{a,s}, \hat{J}^{a,s}(i)$, and $\hat{J}_0^{a,s}(i)$ be defined accordingly. Then the resultant model, denoted as Model **FRR** (fleeting and routing with recapture), is given as follows, where we replace $L^{TF}, A_{TF}, \Pi_{ph}, A, J(i), L^{a,s}, J^{a,s}(i)$, and $J_0^{a,s}(i)$ in the stated equations by the aforementioned corresponding sets designated by $\widehat{(\cdot)}$:

$$\begin{aligned}
\text{FRR :} \quad & \text{Maximize} \quad \sum_{p \in \hat{\Pi}} \sum_{h \in H_p} \sum_{q \in \hat{\Pi}_{ph}} f_{qh} \pi_{pqh} + \sum_{(j,k) \in \hat{A}_{TF}} \sum_{i \in F(j) \cap F(k)} v_{ijk} y_{ijk} - \sum_{j \in \hat{L}} \sum_{i \in F(j)} c_{ij} x_{ij} \\
& - \delta \sum_{p \in \hat{\Pi}} \sum_{h \in H_p} \sum_{q \in \hat{\Pi}_{ph}, q \neq p} \pi_{pqh} + \epsilon_o \sum_{s \in S} \sum_{a \in AT} \sum_{m=1}^{|F_a^s|} \left[w_m \sum_{j \in \hat{L}^{a,s}} j^2 x_{\sigma(m,a,s),j} \right] \quad (5.3.62)
\end{aligned}$$

subject to:

$$\sum_{i \in F(j)} x_{ij} = 1, \quad \forall j \in \hat{L} \quad (5.3.63)$$

$$\sum_{j \in \hat{J}^{a,s}(i)} y_{i,n+s,j} \leq 1, \quad \forall i \in F_a^s, a \in AT, s \in S \quad (5.3.64)$$

$$y_{i,n+s,j} = 0, \quad \forall i \in F_a^s, a \in AT, s \in S, j \in \hat{J}_0^{a,s}(i) \quad (5.3.65)$$

$$\begin{aligned}
& \sum_{i \in F \setminus F^s} \sum_{j: (n+s,j) \in \hat{A}} y_{i,n+s,j} + \sum_{j: (n+s,j) \in \hat{A}} \sum_{i \in F^s \setminus F(j)} y_{i,n+s,j} \\
& + \sum_{j: (j,n+s) \in \hat{A}} \sum_{i \in F \setminus F(j)} y_{i,j,n+s} = 0, \quad \forall s \in S \quad (5.3.66)
\end{aligned}$$

$$\sum_{i \in F_a^s} \sum_{\substack{j \in \hat{J}(i): \\ (n+s,j) \in \hat{A}}} y_{i,n+s,j} = \sum_{i \in F_a} \sum_{\substack{j \in \hat{J}(i): \\ (j,n+s) \in \hat{A}}} y_{i,j,n+s}, \quad \forall s \in S, a \in AT \quad (5.3.67)$$

$$\sum_{k: (j,k) \in \hat{A}, k \in \hat{J}(i)} y_{ijk} = x_{ij}, \quad \forall j \in \hat{L}, i \in F(j) \quad (5.3.68)$$

$$\sum_{k: (k,j) \in \hat{A}, k \in \hat{J}(i)} y_{ikj} = x_{ij}, \quad \forall j \in \hat{L}, i \in F(j) \quad (5.3.69)$$

$$x_{ij} - x_{ik} = 0, \quad \forall i \in F(j) \cap F(k), (j,k) \in \hat{L}^{TF} \quad (5.3.70)$$

$$\begin{aligned}
t_k & \geq t_j - b_j + a_k + \sum_{i \in F(j) \cap F(k)} [\max\{0, t_{ij} + \tau_i + b_j - a_k\} y_{ijk}] \\
& + \sum_{i \in F(k)} \sum_{\substack{h: (h,k) \in \hat{A} \\ h \in \hat{J}(i), h \neq j}} [\max\{0, a_h + t_{ih} + \tau_i - a_k\} y_{ihk}]
\end{aligned}$$

$$+ \sum_{i \in F(j)} \sum_{\substack{l: (j,l) \in \hat{A} \\ l \in \hat{J}(i), l \neq k}} [\max\{0, b_j + t_{ij} + \tau_i - b_l\} y_{ijl}], \quad \forall (j,k) \in \hat{A}, \text{ with } \{j,k\} \subset \hat{L} \quad (5.3.71)$$

$$t_j \geq a_j + \sum_{i \in F(j)} \sum_{\substack{k: (k,j) \in \widehat{A} \\ k \in \widehat{J}(i)}} [\max\{0, a_k + t_{ik} + \tau_i - a_j\} y_{ikj}], \quad \forall j \in \widehat{L} \quad (5.3.72)$$

$$t_j \leq b_j - \sum_{i \in F(j)} \sum_{\substack{k: (j,k) \in \widehat{A} \\ k \in \widehat{J}(i)}} [\max\{0, b_j + t_{ij} + \tau_i - b_k\} y_{ijk}], \quad \forall j \in \widehat{L} \quad (5.3.73)$$

$$\sum_{p \in \widehat{\Pi}} \sum_{q \in \widehat{\Pi}_{ph} \cap \widehat{\Pi}(j)} \pi_{pqh} \leq \sum_{i \in F(j)} Cap_{ih} x_{ij}, \quad \forall j \in \widehat{L}, h \in H \quad (5.3.74)$$

$$\sum_{p \in \widehat{\Pi}(j)} \pi_{pph} \leq \sum_{i \in F(j)} \widetilde{Cap}_{ihj} x_{ij}, \quad \forall j \in \widehat{L}, h \in H \quad (5.3.75)$$

$$\sum_{q \in \widehat{\Pi}_{ph}} \pi_{pqh} \leq \mu_{ph}, \quad \forall p \in \widehat{\Pi}, h \in H_p \quad (5.3.76)$$

$$\pi_{pqh} + \beta_{pqh} \pi_{pph} \leq \beta_{pqh} \mu_{ph}, \quad \forall p \in \widehat{\Pi}, h \in H_p, q \in \widehat{\Pi}_{ph}, q \neq p \quad (5.3.77)$$

$$\sum_{j \in \widehat{L}^{a,s}} j^2 x_{\sigma(m,a,s),j} \geq \sum_{j \in \widehat{L}^{a,s}} j^2 x_{\sigma(m+1,a,s),j}, \quad \forall m = 1, \dots, |F_a^s| - 1, a \in AT, s \in S \quad (5.3.78)$$

$$\sum_{j \in \widehat{J}(i)} t_{ij} x_{ij} + \sum_{s \in S_{NM}} \sum_{\substack{j \in \widehat{J}(i): \\ (j,n+s) \in \widehat{A}}} t_{is} y_{i,j,n+s} \leq \lambda_a, \quad \forall i \in F_a, a \in AT \quad (5.3.79)$$

$$\sum_{i \in F_a} \left[\sum_{j \in \widehat{J}(i)} t_{ij} x_{ij} + \sum_{s \in S_{NM}} \sum_{\substack{j \in \widehat{J}(i): \\ (j,n+s) \in \widehat{A}}} t_{is} y_{i,j,n+s} \right] \leq \lambda_a^{tot}, \quad \forall a \in AT \quad (5.3.80)$$

$$(x, y) : \text{binary}, \quad \pi \geq 0, \quad t : \text{continuous}. \quad (5.3.81)$$

(Step A1.4) Re-optimization Process: Let $\overline{\text{FRR}}$ denote Problem FRR in which we relax the y -variables to be continuous (nonnegative), while enforcing only the x -variables to be binary-valued. We now apply Benders decomposition to Problem $\overline{\text{FRR}}$ by projecting onto the x -variable space in the corresponding master program, where the relevant constraints on the x -variables include (5.3.63), (5.3.10), (5.3.24), and (5.3.81), as well as the implied inequalities obtained by omitting the terms involving the y -variables in (5.3.25) and (5.3.26). Note that these latter constraints, as also the restricted set $F(j)$ in (5.3.63), $\forall j \in \widehat{L}$, help control and guide the x -variables in the master program. Here too, as per Remark 5.2, we generate nondominated or strong Benders cuts using the prescribed perturbation strategy.

We present the Benders master program, subproblem, and the corresponding Benders cuts related to Problem $\overline{\text{FRR}}$, where the variable η_1 represents the value function of the resulting subproblem for any fixed x . This yields the following decomposition:

(Master Program)

$$\widehat{\text{MP}} : \text{ Maximize } - \sum_{j \in \widehat{L}} \sum_{i \in F(j)} c_{ij} x_{ij} + \epsilon_o \sum_{s \in S} \sum_{a \in AT} \sum_{m=1}^{|F_a^s|} \left[w_m \sum_{j \in \widehat{L}^{a,s}} j^2 x_{\sigma(m,a,s),j} \right] + \eta_1$$

subject to:

$$\eta_1 \leq BC(\alpha, u, \sigma, \omega, \rho, \gamma, \tau, \theta, \varphi), \quad \forall (\alpha, u, \sigma, \omega, \rho, \gamma, \tau, \theta, \varphi) \in \widehat{\nabla} \quad (5.3.82)$$

$$\sum_{i \in F(j)} x_{ij} = 1, \quad \forall j \in \widehat{L} \quad (5.3.83)$$

$$x_{ij} - x_{ik} = 0, \quad \forall i \in F(j) \cap F(k), (j, k) \in \widehat{L}^{TF} \quad (5.3.84)$$

$$\sum_{j \in \widehat{L}^{a,s}} j^2 x_{\sigma(m,a,s),j} \geq \sum_{j \in \widehat{L}^{a,s}} j^2 x_{\sigma(m+1,a,s),j}, \quad \forall m = 1, \dots, |F_a^s| - 1, a \in AT, s \in S \quad (5.3.85)$$

$$\sum_{j \in \widehat{J}(i)} t_{ij} x_{ij} \leq \lambda_a, \quad \forall i \in F_a, a \in AT \quad (5.3.86)$$

$$\sum_{i \in F_a} \sum_{j \in \widehat{J}(i)} t_{ij} x_{ij} \leq \lambda_a^{tot}, \quad \forall a \in AT \quad (5.3.87)$$

$$x : \text{ binary}, \quad \eta_1 : \text{ unrestricted}, \quad (5.3.88)$$

where the Benders cuts (5.3.82) are derived from the dual to the subproblem defined below, and where $\widehat{\nabla}$ represents the set of extreme points of the polyhedron defined by the constraints of this dual subproblem. Note also that we have accommodated artificial variables $x^a \equiv (x_{ij}^a, \forall j \in \widehat{L}, i \in F(j))$ within Constraints (5.3.94) and (5.3.95) below along with a suitable penalty term in the objective function (5.3.89) for a sufficiently large $M > 0$, so that the subproblem is feasible and bounded for any fixed \bar{x} , and thus has an optimum (the solution $(y, \pi, x^a) \equiv (0, 0, \bar{x})$ with $a \leq t \leq b$ is feasible).

(Subproblem)

$$\widehat{\text{PSP}}(\bar{x}) : \text{Maximize} \quad \sum_{p \in \widehat{\Pi}} \sum_{h \in H_p} \sum_{q \in \widehat{\Pi}_{ph}} f_{qh} \pi_{pqh} + \sum_{(j,k) \in \widehat{A}_{TF}} \sum_{i \in F(j) \cap F(k)} v_{ijk} y_{ijk} \\ - \delta \sum_{p \in \widehat{\Pi}} \sum_{h \in H_p} \sum_{q \in \widehat{\Pi}_{ph}, q \neq p} \pi_{pqh} - Me^T x^a \quad (5.3.89)$$

subject to:

$$\sum_{j \in \widehat{J}^{a,s}(i)} y_{i,n+s,j} \leq 1, \quad \forall i \in F_a^s, a \in AT, s \in S \quad (\alpha_{ias}^1) \quad (5.3.90)$$

$$y_{i,n+s,j} = 0, \quad \forall i \in F_a^s, a \in AT, s \in S, j \in \widehat{J}_0^{a,s}(i) \quad (\alpha_{iasj}^2) \quad (5.3.91)$$

$$\sum_{i \in F \setminus F^s} \sum_{j: (n+s,j) \in \widehat{A}} y_{i,n+s,j} + \sum_{j: (n+s,j) \in \widehat{A}} \sum_{i \in F^s \setminus F(j)} y_{i,n+s,j} \\ + \sum_{j: (j,n+s) \in \widehat{A}} \sum_{i \in F \setminus F(j)} y_{i,j,n+s} = 0, \quad \forall s \in S \quad (u_s^1) \quad (5.3.92)$$

$$\sum_{i \in F_a^s} \sum_{\substack{j \in \widehat{J}(i): \\ (n+s,j) \in \widehat{A}}} y_{i,n+s,j} - \sum_{i \in F_a} \sum_{\substack{j \in \widehat{J}(i): \\ (j,n+s) \in \widehat{A}}} y_{i,j,n+s} = 0, \quad \forall s \in S, a \in AT \quad (u_{as}^2) \quad (5.3.93)$$

$$\sum_{k: (j,k) \in \widehat{A}, k \in \widehat{J}(i)} y_{ijk} + x_{ij}^a = \bar{x}_{ij}, \quad \forall j \in \widehat{L}, i \in F(j) \quad (\sigma_{ij}^1) \quad (5.3.94)$$

$$\sum_{k: (k,j) \in \widehat{A}, k \in \widehat{J}(i)} y_{ikj} + x_{ij}^a = \bar{x}_{ij}, \quad \forall j \in \widehat{L}, i \in F(j) \quad (\sigma_{ij}^2) \quad (5.3.95)$$

$$t_j - t_k + \sum_{i \in F(j) \cap F(k)} [\max\{0, t_{ij} + \tau_i + b_j - a_k\} y_{ijk}] \\ + \sum_{i \in F(k)} \sum_{\substack{h: (h,k) \in \widehat{A} \\ h \in \widehat{J}(i), h \neq j}} [\max\{0, a_h + t_{ih} + \tau_i - a_k\} y_{ihk}]$$

$$+ \sum_{i \in F(j)} \sum_{\substack{l: (j,l) \in \widehat{A} \\ l \in \widehat{J}(i), l \neq k}} [\max\{0, b_j + t_{ij} + \tau_i - b_l\} y_{ijl}] \leq b_j - a_k,$$

$$\forall (j,k) \in \widehat{A}, \text{ with } \{j,k\} \subset \widehat{L} \quad (\omega_{jk}) \quad (5.3.96)$$

$$\sum_{i \in F(j)} \sum_{\substack{k: (k,j) \in \widehat{A} \\ k \in \widehat{J}(i)}} [\max\{0, a_k + t_{ik} + \tau_i - a_j\} y_{ikj}] - t_j \leq -a_j, \quad \forall j \in \widehat{L} \quad (\rho_j^1) \quad (5.3.97)$$

$$\sum_{i \in F(j)} \sum_{\substack{k: (j,k) \in \widehat{A} \\ k \in \widehat{J}(i)}} [\max\{0, b_j + t_{ij} + \tau_i - b_k\} y_{ijk}] + t_j \leq b_j, \quad \forall j \in \widehat{L} \quad (\rho_j^2) \quad (5.3.98)$$

$$\sum_{p \in \widehat{\Pi}} \sum_{q \in \widehat{\Pi}_{ph} \cap \widehat{\Pi}(j)} \pi_{pqh} \leq \sum_{i \in F(j)} Cap_{ih} \bar{x}_{ij}, \quad \forall j \in \widehat{L}, \forall h \in H \quad (\gamma_{jh}^1) \quad (5.3.99)$$

$$\sum_{p \in \widehat{\Pi}(j)} \pi_{pph} \leq \sum_{i \in F(j)} \widetilde{Cap}_{ihj} \bar{x}_{ij}, \quad \forall j \in \widehat{L}, \forall h \in H \quad (\gamma_{jh}^2) \quad (5.3.100)$$

$$\sum_{q \in \widehat{\Pi}_{ph}} \pi_{pqh} \leq \mu_{ph}, \quad \forall p \in \widehat{\Pi}, h \in H_p \quad (\tau_{ph}) \quad (5.3.101)$$

$$\pi_{pqh} + \beta_{pqh} \pi_{pph} \leq \beta_{pqh} \mu_{ph}, \quad \forall p \in \widehat{\Pi}, h \in H_p, q \in \widehat{\Pi}_{ph}, q \neq p \quad (\theta_{pqh}) \quad (5.3.102)$$

$$\sum_{s \in S_{NM}} \sum_{\substack{j \in \widehat{J}(i): \\ (j, n+s) \in \widehat{A}}} t_{is} y_{i,j, n+s} \leq \lambda_a - \sum_{j \in \widehat{J}(i)} t_{ij} \bar{x}_{ij}, \quad \forall i \in F_a, a \in AT \quad (\varphi_{ia}^1) \quad (5.3.103)$$

$$\sum_{i \in F_a} \sum_{s \in S_{NM}} \sum_{\substack{j \in \widehat{J}(i): \\ (j, n+s) \in \widehat{A}}} t_{is} y_{i,j, n+s} \leq \lambda_a^{tot} - \sum_{i \in F_a} \sum_{j \in \widehat{J}(i)} t_{ij} \bar{x}_{ij}, \quad \forall a \in AT \quad (\varphi_a^2) \quad (5.3.104)$$

$$(y, \pi, x^a) \geq 0, \quad t : \text{continuous}, \quad (5.3.105)$$

where the corresponding dual variables are displayed next to each associated subproblem constraint. Based on the resulting optimal values $(\bar{\alpha}, \bar{u}, \bar{\sigma}, \bar{\omega}, \bar{\rho}, \bar{\gamma}, \bar{\tau}, \bar{\theta}, \bar{\varphi})$ of these dual variables, we derive the corresponding Benders cut (5.3.89) as follows:

$$\begin{aligned} \eta_1 \leq & \sum_{j \in \widehat{L}} \sum_{i \in F(j)} [\bar{\sigma}_{ij}^1 + \bar{\sigma}_{ij}^2] x_{ij} + \sum_{j \in \widehat{L}} \sum_{i \in F(j)} \sum_{h \in H} [Cap_{ih} \bar{\gamma}_{jh}^1 + \widetilde{Cap}_{ihj} \bar{\gamma}_{jh}^2] x_{ij} \\ & - \sum_{a \in AT} \sum_{i \in F_a} \sum_{j \in \widehat{J}(i)} t_{ij} (\bar{\varphi}_{ia}^1 + \bar{\varphi}_a^2) x_{ij} + \sum_{s \in S} \sum_{a \in AT} \sum_{i \in F_a^s} \bar{\alpha}_{ias} + \sum_{\substack{(j,k) \in \widehat{A} \\ \{j,k\} \subset \widehat{L}}} (b_j - a_k) \bar{\omega}_{jk} + \sum_{j \in \widehat{L}} (b_j \bar{\rho}_j^2 - a_j \bar{\rho}_j^1) \\ & + \sum_{p \in \widehat{\Pi}} \sum_{h \in H_p} \mu_{ph} \bar{\tau}_{ph} + \sum_{p \in \widehat{\Pi}} \sum_{h \in H_p} \sum_{\substack{q \in \widehat{\Pi}_{ph} \\ :q \neq p}} \beta_{pqh} \mu_{ph} \bar{\theta}_{pqh} + \sum_{a \in AT} \sum_{i \in F_a} \lambda_a \bar{\varphi}_{ia}^1 + \sum_{a \in AT} \lambda_a^{tot} \bar{\varphi}_a^2. \end{aligned}$$

This Benders process iterates until $\bar{\eta}_1 \leq (1 + \epsilon) \nu[\widehat{\text{PSP}}(\bar{x})]$, with the optimality tolerance set to be $\epsilon = 10^{-4}$, or until a maximum of 30 cuts are generated.

Next, we fix the values of (x, π) as obtained from the resulting solution to $\overline{\text{FRR}}$, and finally solve the residual problem in the variables y and t , with y declared to be binary-valued.

Algorithm A2:

For **Steps A2.1 - A2.3**, we identically follow the respective steps A1.1 - A1.3 of Algorithm A1. However, in lieu of Step A1.4, after trimming off the unprofitable legs and paths, we solve the resultant Model FRR directly via CPLEX in **Step A2.4**. (As an alternative approach for Step 2.4, we will also experiment with using Step A3.4 described below.)

Algorithm A3:

For **Step A3.1**, we adopt the same procedure as in Step A1.1. Next, at **Step A3.2**, we solve the relaxed MIP given by Model $\overline{\text{SFR}}^+$ directly (using CPLEX) without applying Benders decomposition to this problem. In **Step A3.3**, after obtaining the profitable legs and paths via the previous step and fixing the resulting (z, ξ) -variables, we derive the Model FRR similar to Step A1.3. Finally, in **Step A3.4**, we solve Problem $\overline{\text{FRR}}$ using CPLEX in concert with SOS Type I branching applied with respect to the generalized upper bounding constraints $\sum_{i \in F(j)} x_{ij} = 1, \forall j \in \widehat{L}$ (see Jiang and Barnhart 2009, and Chapter 4). Here, upon solving the (current) LP-relaxation, we select a leg $j' \in \widehat{L}$ such that $\bar{x}_{ij'}$ is fractional for some set of aircraft $i \in F(j')$, where \bar{x} denotes the LP relaxation solution at hand. Next, we extract two indices i_1 and i_2 for the corresponding two largest fractional variables $x_{i_1 j'}$ and $x_{i_2 j'}$, and assign these two aircraft to the sets B_1 and B_2 , respectively. For the remaining aircraft, suppose that the indices $i \in F(j')$ are ordered according to nonincreasing values of $Cap_i \equiv \sum_{h \in H} Cap_{ih}$, and define $\overline{Cap} \equiv \sum_{i \in F(j')} Cap_i \bar{x}_{ij'}$. Then we partition these remaining aircraft indices into $\{i \in F(j') \setminus \{i_1, i_2\} : Cap_i \geq \overline{Cap}\}$ and $\{i \in F(j') \setminus \{i_1, i_2\} : Cap_i < \overline{Cap}\}$, and append the latter two sets to B_1 and B_2 , respectively. Accordingly, we then branch on the dichotomy: $\{\sum_{i \in B_1} x_{ij'} = 1\} \vee \{\sum_{i \in B_2} x_{ij'} = 1\}$. Having thus obtained a solution to Problem $\overline{\text{FRR}}$, we follow a process similar to Step A1.4 to derive the final complete prescribed solution by fixing the resulting values of (x, π) and then using CPLEX to solve the residual problem in (y, t) with y declared to be binary-valued.

5.4 Computational Results

In this section, we provide computational results for several sets of experiments. For the first set presented in Section 5.4.1, as a preliminary step, we solve Model SFR⁺ using the different algorithmic variants described in Section 5.3 and identify the most effective strategy among them. Next, using the best algorithm from Section 5.4.1, we conduct the following sensitivity analyses in Section 5.4.2. First, we explore the costs and benefits of using different sets of through-flights, and then, we study the effect of varying the individual parameters related to the symmetry-breaking constraints and the effect of the demand recapture feature, respectively, and we analyze their corresponding impacts.

Five test instances based on real data provided by *United Airlines* were generated for experimental purposes as follows: T1 (280 flights, 829 paths), T2 (342 flights, 1202 paths), T3 (466 flights, 2121 paths), T4 (528 flights, 2689 paths), and T5 (592 flights, 3342 paths), where the proportion of non-stop, two-leg, and three-leg itineraries were about 20.6%, 70.5%, and 8.9%, respectively, on average. To enhance scalability and feasibility, the data sets were generated by strategically positioning aircraft of different types at their initial stations at the beginning of the scheduling horizon (according to what might be expected in practice), as well as by keeping the number of aircraft for each type in proportion to the number of legs in each test instance. All runs were made using the commercial software OPL CPLEX 10.1 on a Dimension 8400 computer having an Intel Pentium 2.99 GHz processor, with 2.50 GB of RAM, and running Windows XP. The quality of the solution attained for each experiment is reported in relative terms for comparative purposes. We define **% gap** as $100(F^* - F)/F^*$, where F and F^* respectively denote the objective value attained by a given method and the best objective value found across all procedures. Furthermore, for use in Section 5.4.2, we define **% Improvement** as $100(G - G_0)/G_0$, where G_0 and G respectively denote the objective value obtained for the baseline case and the objective value attained for a given varied case. The computational effort required by any procedure or test case scenario is specified in CPU hours, where the run-time limit was set as 12 CPU hours. A default

optimality tolerance of $\epsilon = 10^{-6}$ was used for all runs.

5.4.1 Preliminary Test of Algorithms

To measure the effectiveness of utilizing the Benders decomposition approach as well as that of the SOS Type I branching procedure, we tested Algorithms A1, A2, and A3 on the baseline Model SFR⁺. Here, as specified in Section 5.3, Algorithm A2 is further classified as A2' and A2'', where we solve Problem FRR in Step A2.4 directly via CPLEX in Algorithm A2', and via Step A3.4 in Algorithm A2''. The valid inequalities (5.2.40) within Model SFR⁺ were generated by identifying the indices j^* that yielded the five most violated inequalities with respect to the LP relaxation solution, out of up to 20 potential candidates identified using a circular list, where this cut generation process was repeated for up to three rounds. Also, the Benders iterative processes within Algorithms A1 and A2 were set to generate a maximum of 30 (nondominated) Benders cuts. In addition, to be consistent across all runs in this experiment, we used the following default settings of parameters: (i) the weights corresponding to the different terms within the symmetry-breaking constraints were computed using $a = 1$, $b = 1$, and $c = 0$; and (ii) the objective penalty parameter δ for recapture consideration or re-booking cost was fixed at \$100 per passenger.

Table 5.1 presents the results obtained. For the largest test instance T5, all the algorithms terminated at the 12 hour CPU time limit due to the more intricate aircraft assignment decisions and routing connection choices, thus indicating greater computational requirements for this data size. As a point of interest in regard to the convergence of Algorithm A1, we ran the test instance T5 using this procedure for extended computational run time limits of 18 hours and 24 hours. Incrementally, the objective function value improved by 0.34% for the 18 hour run time limit, and it further improved by 0.06% when this limit was extended to 24 hours, which respectively correspond to improved expected annual profits of \$1.3M and \$0.2M. Algorithm A3, which relies directly on CPLEX to solve the problem, terminated at the 12 hour CPU time limit for all test cases, where for the two largest instances T4 and

Table 5.1: Comparative Results for Algorithms A1, A2, and A3

| Test Instance | Algorithms | | | |
|------------------------|-------------|-------------|-------------|-------------|
| | A1 | A2' | A2'' | A3 |
| T1: | | | | |
| % gap | 0.00 | 0.48 | 0.21 | 1.03 |
| CPU Time (hrs.) | 8.0 | 12.0 | 9.6 | 12.0 |
| T2: | | | | |
| % gap | 0.16 | 0.51 | 0.00 | 2.37 |
| CPU Time (hrs.) | 9.8 | 12.0 | 10.2 | 12.0 |
| T3: | | | | |
| % gap | 0.00 | 1.22 | 0.53 | 4.01 |
| CPU Time (hrs.) | 10.4 | 12.0 | 11.2 | 12.0 |
| T4: | | | | |
| % gap | 0.00 | 0.79 | 0.60 | * |
| CPU Time (hrs.) | 11.3 | 12.0 | 12.0 | 12.0 |
| T5: | | | | |
| % gap | 0.00 | 2.87 | 1.19 | * |
| CPU Time (hrs.) | 12.0 | 12.0 | 12.0 | 12.0 |
| Average: | | | | |
| % gap | 0.03 | 1.17 | 0.51 | 2.47 |
| CPU Time (hrs.) | 10.3 | 12.0 | 11.0 | 12.0 |

* No solution found.

T5, this method was unable to find even a feasible solution in the set time limit. This clearly demonstrates that applying Benders decomposition has an overall advantage of obtaining solutions relatively faster. The average CPU times required by Algorithms A1, A2', A2'', and A3 were 10.3, 12, 11, and 12 hours, respectively. Not only did Algorithm A1 consume the least CPU time, but also, overall, it achieved the best objective function value, where the average % gaps realized by A2', A2'', and A3 relative to A1 were 1.14%, 0.48%, and 2.44%, respectively, which correspondingly translate to reduced expected annual profit values of

\$4.4M, \$1.9M, and \$9.5M. When comparing the performances of the two algorithmic variants A2' and A2'' of A2, we observe that A2'' outperformed A2' by achieving a % gap of 0.51% versus 1.17%, and moreover, required less computational effort on average by about one CPU hour, which further indicates the advantage of using the proposed SOS Type I branching strategy rather than relying on CPLEX alone to make branching decisions in the final step of Algorithm A2.

We select the best performing procedure, Algorithm A1, for henceforth conducting the different sensitivity analyses in the following sections in order to study the effects of various modeling features.

5.4.2 Analyses of Different Modeling Features

Chapters 3 and 4 have already demonstrated that integrating schedule design and fleet assignment yields greater profits in comparison with a sequential schedule design and fleet assignment approach. Hence, in this section, we focus on assessing the separate effects of incorporating the particular modeling features of through-flights, symmetry-breaking constraints, and demand recapture consideration. For each feature in turn, we provide insights into its benefits by conducting a sensitivity analysis with respect to its relevant parameters, while fixing elements pertaining to the other features. The following sections describe the experiments and analyses for the foregoing three cases, respectively, and then we exhibit the benefits of using the proposed valid inequalities at the end.

Impact of Through-flights

In this experiment, we examine how varying the relative proportion of through-flights impacts the total net profit. For each test instance (T1-T5), three different sets of through-flights were generated by constructing different combinations of possible routing connections, based on the length of the time interval between two sequential flight legs being between 30 and

35 minutes, between 30 and 40 minutes, and between 30 and 45 minutes. These cases are respectively denoted as L_1^{TF} , L_2^{TF} , and L_3^{TF} . For comparison purposes, we also define a *baseline case* L_0^{TF} for which $L^{TF} = L_0^{TF} \equiv \phi$. For each of these cases, we initially let $A_{TF} = \phi$, and we refer to the corresponding scenarios as $L_r^{TF}(A_{TF} = \phi)$, $r = 0, 1, 2$, and 3. Considering that a pair of through-flight legs on any given path has an effect on the demand distribution for that path, we adjusted the corresponding itinerary demands in order to render the scenarios more practical. More specifically, we identified the set of paths containing each mandatory through-flight pair, and increased the corresponding demands on these paths (randomly) by 15-20%. (On average, the number of paths per through-flight pair is 1.6.) Furthermore, in a separate set of runs, we examined the cases of L_0^{TF} , L_1^{TF} , and L_2^{TF} , and holding the demand structure the same as before for each of these cases, we provided an additional opportunity for enhanced revenue by letting $A_{TF} = A_{TF}^r \equiv L_3^{TF} \setminus L_r^{TF}$ for $r = 0, 1$, and 2, respectively. Accordingly, the v_{ijk} -parameters were determined as 20% of the average lowest class fare over all the paths containing legs j and k when served by aircraft i , $\forall (j, k) \in A_{TF}$, $i \in F(j) \cap F(k)$. We refer to these respective cases as $L_r^{TF}(A_{TF} = A_{TF}^r)$, for $r = 0, 1$, and 2. Note that the mandatory through-flights are ascertained based on market considerations and are assumed to influence demands and thereby provide the opportunity for increased revenues depending on available capacity, whereas the optional through-flights provide the opportunity of increased revenues through higher fares that potential passengers are willing to pay for the added convenience.

We treat $L_0^{TF}(A_{TF} = \phi)$ as a baseline case in Table 5.2, and compare the results of the other cases against it. The results in Table 5.2 indicate that, within the cases of L_r^{TF} for $r = 0, 1$, and 2, $L_r^{TF}(A_{TF} = A_{TF}^r)$ performed better than $L_r^{TF}(A_{TF} = \phi)$, achieving a higher objective value on average by 1.61%, 1.47%, and 0.94%, respectively, but consuming additional computational effort. Having extra through-flight legs modeled as additional revenue sources resulted in improved expected annual profits of \$2.3M, \$2.1M, and \$1.4M, respectively, at the modest expense of increasing computational times by 9.3%, 5.6%, and 3.2%, respectively. When comparing the results for $L_0^{TF}(A_{TF} = \phi)$ with those for $L_1^{TF}(A_{TF} =$

Table 5.2: Effect of Accommodating Through-flights

| Test Instance | L_0^{TF} | | L_1^{TF} | | L_2^{TF} | | L_3^{TF} |
|-----------------|-----------------|---------------------|-----------------|---------------------|-----------------|---------------------|-----------------|
| | $A_{TF} = \phi$ | $A_{TF} = A_{TF}^0$ | $A_{TF} = \phi$ | $A_{TF} = A_{TF}^1$ | $A_{TF} = \phi$ | $A_{TF} = A_{TF}^2$ | $A_{TF} = \phi$ |
| T1: | | | | | | | |
| % Improvement | 0.00 | 1.03 | 0.11 | 1.15 | 0.22 | 0.55 | 0.17 |
| CPU Time (hrs.) | 7.9 | 9.4 | 8.0 | 8.8 | 8.0 | 8.5 | 8.2 |
| T2: | | | | | | | |
| % Improvement | 0.00 | 1.91 | 0.17 | 1.80 | 0.25 | 1.31 | 0.33 |
| CPU Time (hrs.) | 9.8 | 10.8 | 9.8 | 10.3 | 10.0 | 10.2 | 10.0 |
| T3: | | | | | | | |
| % Improvement | 0.00 | 0.80 | 0.13 | 0.74 | 0.08 | 0.42 | 0.27 |
| CPU Time (hrs.) | 10.3 | 12.0 | 10.4 | 11.4 | 10.6 | 10.9 | 10.8 |
| T4: | | | | | | | |
| % Improvement | 0.00 | 2.97 | 0.37 | 3.08 | 0.28 | 2.82 | 0.51 |
| CPU Time (hrs.) | 11.3 | 12.0 | 11.3 | 12.0 | 11.4 | 12.0 | 11.6 |
| T5: | | | | | | | |
| % Improvement | 0.00 | 1.34 | 0.09 | 1.41 | 0.14 | 0.56 | 0.18 |
| CPU Time (hrs.) | 12.0 | 12.0 | 12.0 | 12.0 | 12.0 | 12.0 | 12.0 |
| Average: | | | | | | | |
| % Improvement | 0.00 | 1.61 | 0.17 | 1.64 | 0.19 | 1.13 | 0.29 |
| CPU Time (hrs.) | 10.3 | 11.2 | 10.3 | 10.9 | 10.4 | 10.7 | 10.5 |

ϕ), $L_2^{TF}(A_{TF} = \phi)$, and $L_3^{TF}(A_{TF} = \phi)$, on average, the latter three cases yielded improved solutions with enhanced profits of 0.17%, 0.19%, and 0.29%. On the other hand, for the cases of $L_r^{TF}(A_{TF} = A_{TF}^r)$, for $r = 0, 1$, and 2 , the best objective value was obtained by $L_1^{TF}(A_{TF} = A_{TF}^1)$, which improved profits over the cases of $L_0^{TF}(A_{TF} = A_{TF}^0)$ and $L_2^{TF}(A_{TF} = A_{TF}^2)$ by 0.03% and 0.48%, respectively. This suggests that composing the right-mix of both forced and flexible through-flight leg pairs in the model, depending on their relative effects on demands and enhanced revenues, can critically influence profit margins. This is relevant to yield management considerations, in that, by providing better service to passengers via appropriate through-flights, airlines can attract more demand that they can attempt to capture by assigning increased seat capacities on appropriate legs, or they can

induce passengers to accept paying higher fares on certain leg segments, or both, in order to enhance revenues. Overall, $L_1^{TF}(A_{TF} = A_{TF}^1)$ achieved the best solution, with a 1.64% improvement over the baseline case of $L_0^{TF}(A_{TF} = \phi)$, which translates to an estimated increase in expected annual profits of \$2.4 million.

Symmetry-breaking Constraints

As discussed in Section 5.2.1, hierarchical symmetry-breaking constraints assign specific identities to indistinguishable variables. More specifically, hierarchical constraints of the type (5.2.20) impose ordered restrictions that can reduce the search domain of the solution space by eliminating symmetric feasible solutions that are not consistent with the specified hierarchy. While introducing such symmetry-breaking restrictions within the set of constraints, we also suitably perturb the original objective function with a weighted sum of terms derived from the corresponding hierarchical constraints as in Ghoniem and Sherali (2009). For this experiment, we consider the following variations: (a) symmetry-breaking constraints alone, versus (b) symmetry-breaking constraints combined with a perturbed objective function; where for the symmetry-breaking constraint (5.2.20), we utilize (c) a coefficient of j^2 for each hierarchical term (as stated in (5.2.20)), or (d) a coefficient of j for each hierarchical term. This yields four different combinations of experimental settings given by (a, c), (a, d), (b, c), and (b, d), which are respectively denoted as SD^1 , SD^2 , SD^3 , and SD^4 . We also denote the case SD^0 as a baseline case in this experiment, where no symmetry-breaking feature is incorporated within the model formulation.

Table 5.3 presents the results obtained. First of all, comparing the case SD^1 with the case SD^2 where the coefficient j^2 in (5.2.20) is replaced by j , the results indicate that there is no discernable difference between these two formulations; both produced similar solutions with the same computational effort of about 11.5 CPU hours on average. Next, as compared with the baseline case, applying symmetry-breaking constraints alone as in option (a) (SD^1 and SD^2), improved profits by 2.21% on average, and also reduced the required computa-

Table 5.3: Computational Results with Different Symmetry-breaking Constraint Settings

| Test Instance | Symmetry-breaking Constraint Settings | | | | |
|------------------------|---------------------------------------|------------------------|------------------------|------------------------|------------------------|
| | SD ⁰ | SD ¹ (a, c) | SD ² (a, d) | SD ³ (b, c) | SD ⁴ (b, d) |
| T1: | | | | | |
| % Improvement | 0.00 | 2.71 | 2.71 | 2.71 | 2.71 |
| CPU Time (hrs.) | 12.0 | 9.5 | 9.5 | 8.0 | 7.8 |
| T2: | | | | | |
| % Improvement | 0.00 | 1.68 | 1.68 | 1.91 | 1.91 |
| CPU Time (hrs.) | 12.0 | 12.0 | 12.0 | 9.8 | 9.7 |
| T3: | | | | | |
| % Improvement | 0.00 | 1.94 | 1.94 | 2.78 | 2.78 |
| CPU Time (hrs.) | 12.0 | 12.0 | 12.0 | 10.4 | 10.1 |
| T4: | | | | | |
| % Improvement | 0.00 | 2.16 | 2.16 | 3.93 | 3.93 |
| CPU Time (hrs.) | 12.0 | 12.0 | 12.0 | 11.3 | 10.9 |
| T5: | | | | | |
| % Improvement | 0.00 | 2.57 | 2.58 | 3.73 | 3.73 |
| CPU Time (hrs.) | 12.0 | 12.0 | 12.0 | 12.0 | 12.0 |
| Average: | | | | | |
| % Improvement | 0.00 | 2.21 | 2.21 | 3.01 | 3.01 |
| CPU Time (hrs.) | 12.0 | 11.5 | 11.5 | 10.3 | 10.1 |

tional effort by 4% on average. When the symmetry-breaking constraints were combined with the perturbation terms in the objective function as in option (b) (SD³ and SD⁴), this further improved the objective function value by 0.8%, which yields an estimated increase in annual profits of \$5.8M over the baseline case, while simultaneously requiring lesser computational effort by 14% on average as compared with the baseline case. This indicates that the symmetric combinatorial nature of the problem significantly benefits from incorporating symmetry-breaking constraints in the model formulation, combined with the objective perturbation terms that guide the underlying LP solutions to more readily conform with the hierarchy imposed by the symmetry-breaking constraints. Overall, the simple coefficient structure (j in lieu of j^2) within the symmetry-breaking constraints (5.2.20) along with the objective perturbation terms (Strategy SD⁴) performed best, and we shall henceforth adopt

this strategy.

Recapture Effect

For recapture consideration, we examined about 50% of the OD markets and identified a set of compatible itineraries within each OD market, and accordingly generated the recapture rates β_{pqh} using the Quality of Service Index (QSI) data for each relevant itinerary. In conducting a sensitivity analysis related to re-accommodating passengers spilled from their original itinerary choices, the penalty parameter value δ was varied as 1, 50, and 100 dollars for each re-routed passenger. For each δ -value, we computed $\% \text{Recap}$ as $100(\sum_{p \in \Pi} \sum_{h \in H_p} \sum_{q \in \Pi_{ph}, q \neq p} \pi_{pqh}) / (\sum_{p \in \Pi} \sum_{h \in H_p} \mu_{ph})$ to assess the proportion of demand that is recaptured on alternative compatible routes in the resulting solution.

Table 5.4 compares the results for different values of the demand recapture penalty parameter δ , using the *No Recapture* scenario as a baseline case. As δ decreases, more recapture occurs among compatible itineraries, and we progressively obtain greater improvements in profits. As compared with the baseline, the cases $\delta = 100, 50,$ and 1 (\$/PAX) achieved higher profits by 1.90%, 1.99%, and 2.10%, respectively, which equivalently translate to estimated increases in annual profits of \$3.7M, \$3.8M, and \$4.0M. Each case of $\delta = 100, 50,$ and 1 (\$/PAX) respectively required, on average, 10.3, 10.5, and 10.8 CPU hours to solve, which indicates an increase in computational effort as δ decreases due to a greater extent of demand switching itineraries rather than simply being spilled. We observe that, for $\delta = 100, 50,$ and 1 (\$/PAX), about 3.2%, 3.4%, and 3.7% of the original demand was respectively re-accommodated among alternative compatible itineraries, exhibiting a limiting marginal recapture rate as the penalty parameter approaches zero, where $\delta = 0$ would essentially permit any possible and profitable recapture occurrence without penalty. In addition, as a point of interest, we assessed the load factor for each case. While the average load factor for the baseline case was 80.4%, that of the cases with $\delta = 100, 50,$ and 1 (\$/PAX) was 83.1%, 83.6%, and 84.5%, respectively, which reflects the higher seat occupancy rates as

Table 5.4: Computational Results with Recapture Consideration

| Test Instance | No Recapture | Re-booking Cost Parameter δ (\$/PAX) | | |
|------------------------|-----------------|---|-------------|-------------|
| | | 100 | 50 | 1 |
| T1: | | | | |
| % Improvement | 0.00 | 1.58 | 1.58 | 1.70 |
| CPU Time (hrs.) | 7.0 | 8.0 | 8.2 | 8.4 |
| % Recap | N/A | 2.4 | 2.4 | 2.6 |
| T2: | | | | |
| % Improvement | 0.00 | 0.76 | 0.82 | 0.89 |
| CPU Time (hrs.) | 7.3 | 9.8 | 10.2 | 10.3 |
| % Recap | N/A | 3.0 | 3.1 | 3.5 |
| T3: | | | | |
| % Improvement | 0.00 | 2.42 | 2.56 | 2.68 |
| CPU Time (hrs.) | 9.0 | 10.4 | 10.4 | 11.2 |
| % Recap | N/A | 2.7 | 3.0 | 3.2 |
| T4: | | | | |
| % Improvement | 0.00 | 1.94 | 2.05 | 2.23 |
| CPU Time (hrs.) | 9.5 | 11.3 | 11.5 | 12.0 |
| % Recap | N/A | 4.7 | 4.9 | 5.2 |
| T5: | | | | |
| % Improvement | 0.00 | 2.81 | 2.95 | 3.01 |
| CPU Time (hrs.) | 12.0 | 12.0 | 12.0 | 12.0 |
| % Recap | N/A | 3.4 | 3.5 | 3.9 |
| Average: | | | | |
| % Improvement | 0.00 | 1.90 | 1.99 | 2.10 |
| CPU Time (hrs.) | 9.0 | 10.3 | 10.5 | 10.8 |
| % Recap | N/A | 3.2 | 3.4 | 3.7 |

more spilled demand is recaptured with a decrease in the penalty parameter δ .

Effect of Valid Inequalities

For the purpose of verifying the effectiveness of employing the proposed valid inequalities described in Section 5.2.3, we conducted experiments to compare the results obtained with and without these inequalities. For the latter base case, we suppressed the model enhancements based on Propositions 5.1, 5.2, 5.4, 5.5, and 5.6 by reverting to the original corresponding constraints of Model SFR, and we then applied the overall best performing algorithm selected from Section 5.4.1 to the resulting model, i.e., Algorithm A1. To be more specific, in Step A1.1, we removed all valid inequalities (5.2.26), (5.2.27), (5.2.32), and (5.2.38)-(5.2.40), but not (5.2.37), where the latter are necessary to linearize the model in order to solve it using the ensuing steps.

For the test instance T1, both cases reached the same objective value within the 12 hour run-time limit; however, when we incorporated the aforementioned valid inequalities, the CPU time required for achieving this solution was reduced by 14%. For the remaining instances T2-T5, the effort consumed by the model run without utilizing the proposed valid inequalities exceeded the 12 hour CPU time limit, whereas the average CPU time corresponding to the runs that incorporated these valid inequalities was about 10.9 hours. Moreover, implementing the valid inequalities resulted in a 0.70% improvement in profits over the base case on average, which translates to an increase of \$1.4M in the estimated annual profits. This demonstrates the benefits of tightening the model representation using the proposed valid inequalities while applying Algorithm A1.

Table 5.5: Comparative Results with and without the Proposed Valid Inequalities

| Test Instance | w/o Valid Inequalities | w/ Valid Inequalities |
|------------------------|------------------------|-----------------------|
| T1: | | |
| % Improvement | 0.00 | 0.00 |
| CPU Time (hrs.) | 9.7 | 8.0 |
| T2: | | |
| % Improvement | 0.00 | 0.46 |
| CPU Time (hrs.) | 12.0 | 9.8 |
| T3: | | |
| % Improvement | 0.00 | 1.35 |
| CPU Time (hrs.) | 12.0 | 10.4 |
| T4: | | |
| % Improvement | 0.00 | 0.72 |
| CPU Time (hrs.) | 12.0 | 11.3 |
| T5: | | |
| % Improvement | 0.00 | 0.95 |
| CPU Time (hrs.) | 12.0 | 12.0 |
| Average: | | |
| % Improvement | 0.00 | 0.70 |
| CPU Time (hrs.) | 11.5 | 10.3 |

Chapter 6

An Integrated Fleet Assignment, Aircraft Routing, and Crew Scheduling Model

6.1 Introduction

In this chapter, we propose a fourth more comprehensive model in which the crew scheduling process is additionally integrated with fleet assignment and aircraft routing. The motivation for this further integration is that crews are qualified to typically serve a set of aircraft types belonging to the same aircraft family, and therefore, the aircraft fleet assignment and routing decisions impact on the decision of assigning crews to flights. Therefore, a new model for the integrated fleet assignment, aircraft routing, and crew scheduling problem is developed to simultaneously handle these important inter-dependent processes. In addition, the model is extended to accommodate various work rules such as imposing a specified minimum and maximum number of flying hours for crews on any given pairing, and a minimum number of departures at a given crew base for each fleet group.

6.2 Model Description and Notation

Crew costs represent the second largest component of airline operating expenses (the largest being fuel costs), easily exceeding a billion dollars annually for large airlines (Bard and Barnhart, 2006). In Model 4, the crew scheduling problem is additionally integrated with fleet assignment and aircraft routing. This integration is crucial for airlines, not only because of the high cost of flight crews, but also because the assignment and routing of aircraft and the assignment of crews are two closely interacting components of the planning process. Indeed, given that a crew of pilots (or *cockpit crew*) is typically qualified to fly only a specified aircraft family, the aircraft fleet and routing decisions impact on the decision of assigning cockpit crews to flights. Moreover, the cost of crew schedules depends not only on the total flight time, but also on the waiting time during connections, as well as on related accommodation expenses (transportation, meals, hotel rooms, etc.) when overnight connections take place outside the crew base. We therefore propose a new model for the integrated fleet, aircraft routing, and crew scheduling problem where all of these important inter-dependent processes are handled simultaneously, and relegate the investigation and testing of suitable solution procedures for future research.

In what follows, a *duty period* is a single workday of a crew, which involves a sequence of flight legs having short rest periods, and also includes briefing and debriefing periods at the beginning and end of the period. A *pairing* is a sequence of duty periods with overnight rests between consecutive duty periods. Each pairing begins and ends at the same station (which is designated as the crew base).

The common practice in most airlines is to first solve the FAP, and then solve a crew scheduling problem (CSP) for each aircraft type, because crews are qualified to serve only one aircraft family. The input of the CSP is a set of flight legs with corresponding departure and arrival times and stations, and the output prescribes a set of duty periods and associated pairings to implement. Because of the restrictive work rules established by the *Federal Aviation Administration* (FAA) and union contracts, constructing optimal legal duties and

pairings is a highly complex problem. Moreover, even when the crew scheduling problem is solved exactly, it results in an overall suboptimal solution since the decisions of assigning aircraft and crews to the flight schedule and that of routing aircraft have been treated separately.

For the sake of tractability, due to the complexity and size of the problem, we do not consider itinerary-based demands in Model 4, but instead consider the traditional leg-based demands. Hence, the costs involve fixed operating costs and the costs of losing spilled passengers due to capacity constraints. More specifically, we assume that there exists a demand for each fare-class h on leg j , and spill occurs when these fare-class-based leg demands are greater than the capacity of the aircraft i assigned to leg j .

Below we define the notation used and provide a description of the underlying network, and then present the basic model formulation.

Sets:

AT : set of aircraft types, indexed by a .

F : set of aircraft, indexed by i .

$F_a \subseteq F$: set of aircraft of type $a \in AT$.

S : set of stations, indexed by s .

$B \subseteq S$: set of crew bases, indexed by s .

$F^s \subseteq F$: set of aircraft whose initial station is $s \in S$.

$F_a^s \subseteq F$: set of aircraft of type $a \in AT$ whose initial station is $s \in S$.

$F_f \subseteq F$: subset of aircraft belonging to aircraft family f ($f = 1, \dots, \varphi$).

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

$nl = |L|$: total number of legs.

D : set of valid duty periods indexed by d .

P : set of valid pairings indexed by p .

$L_p \subseteq L$: set of legs covered by pairing p , $p \in P$.

$D_p \subseteq D$: set of duty periods covered by pairing p , $p \in P$.

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

$H_j \subseteq H$: set of all fare-classes on leg j , $j \in L$; indexed by h .

Parameters:

b_d : cost of duty period d , $d \in D$.

\hat{c}_p : excess cost of pairing p (that is, the difference between the cost of the pairing p and the sum of the costs of the duty periods that belong to D_p), $p \in P$.

c_{ij} : cost of assigning aircraft i to leg j , which includes fixed operating charges (note that since we are not considering optional flights, the leg's demand is the aggregated demand of all itineraries that include this leg).

\tilde{c}_{ijh} : cost of losing spilled passengers based on the excess of the leg demand over the capacity of aircraft i by fare-class h , $i \in F, j \in L, h \in H_j$ (we denote $\tilde{c}_{ij} \equiv \sum_{h \in H_j} \tilde{c}_{ijh}, \forall i \in F, j \in L$).

θ_{fj} : total flying time for leg j when flown by aircraft belonging to family f ($f = 1, \dots, \varphi$), $j \in L$.

τ_i : turn-time of aircraft i , $i \in F$.

$[a_j, b_j]$: departure time-window of leg j , $j \in L$.

t_{ij} : flying duration of leg j when it is assigned to aircraft i , $j \in L, i \in F$.

α_f : lower bound on the total flying time of crews for each pairing when flown by aircraft of family f ($f = 1, \dots, \varphi$).

β_f : upper bound on the total flying time of crews for each pairing when flown by aircraft of family f ($f = 1, \dots, \varphi$).

ρ_{fs} : minimum number of departures at a given crew base s for each family f ($f = 1, \dots, \varphi$), $s \in B$.

δ_{ps} : number of outbound legs at base s that are covered by pairing p , $s \in B, p \in P$.

Underlying Network:

We consider the digraph $G = (V, A)$ where the set of nodes V and the set of arcs A are defined as follows. Each leg $j \in L$ is represented by a node j ($j = 1, \dots, nl$). Furthermore, nodes $nl + s$ ($s \in S$) are added to represent the different stations. The set of arcs A contains three types of arcs:

- $(nl + s, j) \in A \Leftrightarrow$ station s ($s \in S$) is the departure station of flight $j \in L$;
- $(j, nl + s) \in A \Leftrightarrow$ station s ($s \in S$) is the final station of flight $j \in L$;
- $(j, k) \in A \Leftrightarrow$ an aircraft $i \in F$ can cover leg k , immediately after leg j ($j \neq k \in L$).

Decision Variables:

$$x_{ij} : \begin{cases} 1, & \text{if aircraft } i \text{ covers leg } j, i \in F, j \in L \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{ijk} : \begin{cases} 1, & \text{if aircraft } i \text{ serves leg } k \text{ immediately after leg } j, \forall (j, k) \in A \\ & \text{(note that the nodes } j \text{ and } k \text{ could also correspond} \\ & \text{to start and end stations, respectively).} \\ 0, & \text{otherwise.} \end{cases}$$

t_j : departure time of flight $j \in L$.

u_{fp} : binary variable that takes the value 1 if aircraft family f ($f = 1, \dots, \varphi$) is assigned to the legs of pairing $p \in P$, and 0 otherwise.

w_d : binary variable that takes the value 1 if duty period $d \in D$ is selected, and 0 otherwise.

z_p : binary variable that takes the value 1 if pairing $p \in P$ is selected, and 0 otherwise.

Model Formulation:

We present below the *fleeting, routing, and crew scheduling model (FRC)*, which integrates the fleet assignment, aircraft maintenance routing, and crew scheduling processes.

$$\mathbf{FRC:} \quad \text{Minimize} \quad \sum_{i \in F} \sum_{j \in L} (c_{ij} + \tilde{c}_{ij}) x_{ij} + \sum_{d \in D} b_d w_d + \sum_{p \in P} \hat{c}_p z_p \quad (6.2.1)$$

subject to:

$$\sum_{i \in F} x_{ij} = 1, \quad \forall j \in L \quad (6.2.2)$$

$$\sum_{d \in D: j \in L_d} w_d = 1, \quad \forall j \in L \quad (6.2.3)$$

$$\sum_{p \in P: d \in D_p} z_p = w_d, \quad \forall d \in D \quad (6.2.4)$$

$$\sum_{f=1}^{\varphi} u_{fp} = z_p, \quad \forall p \in P \quad (6.2.5)$$

$$\sum_{i \in F_f} x_{ij} = \sum_{p \in P: j \in L_p} u_{fp}, \quad \forall j \in L, \quad f = 1, \dots, \varphi \quad (6.2.6)$$

$$\sum_{j: (nl+s, j) \in A} y_{i, nl+s, j} \leq 1, \quad \forall i \in F^s, \forall s \in S \quad (6.2.7)$$

$$\sum_{i \in F \setminus F^s} \sum_{j: (nl+s, j) \in A} y_{i, nl+s, j} = 0, \quad \forall s \in S \quad (6.2.8)$$

$$\sum_{i \in F_a^s} \sum_{j: (nl+s, j) \in A} y_{i, nl+s, j} = \sum_{i \in F_a} \sum_{j: (j, nl+s) \in A} y_{i, j, nl+s}, \quad \forall s \in S, \forall a \in AT \quad (6.2.9)$$

$$\sum_{k: (j, k) \in A} y_{ijk} = x_{ij}, \quad \forall j \in L, \forall i \in F \quad (6.2.10)$$

$$\sum_{k: (k, j) \in A} y_{ikj} = x_{ij}, \quad \forall j \in L, \forall i \in F \quad (6.2.11)$$

$$(1 - y_{ijk})(t_{ij} + \tau_i + b_j - a_k) \geq t_j + t_{ij} + \tau_i - t_k, \quad \forall i \in F, (j, k) \in A,$$

$$\text{with } \{j, k\} \subset L \quad (6.2.12)$$

$$a_j \leq t_j \leq b_j, \quad \forall j \in L \quad (6.2.13)$$

$$u, x, y, z, w \text{ are binary variables, } \quad t \text{ continuous (see (6.2.13)).} \quad (6.2.14)$$

The objective function (6.2.1) seeks to minimize the total fleet assignment (including fixed operating and spill) costs plus the crew scheduling costs. Constraint (6.2.2) is the coverage constraint that requires the assignment of an aircraft to each leg, and Constraint (6.2.3) requires that each leg must be covered by exactly one duty period. Constraint (6.2.4) enforces that each selected duty period must be covered by exactly one pairing, and also ensures that if $z_p = 1$, then we also have $w_d = 1, \forall d \in D_p$. Constraint (6.2.5) requires that each selected pairing must be flown by exactly one aircraft family, while Constraint (6.2.6) asserts that if an aircraft from family f is assigned to a leg j , then the pairing containing this leg must also have the same family f allocated to it. In effect, note that for any $\hat{p} \in P, f = 1, \dots, \varphi$, by summing (6.2.6) over $j \in L_{\hat{p}}$, we get that $\sum_{i \in F_f} \sum_{j \in L_{\hat{p}}} x_{ij} = \sum_{j \in L_{\hat{p}}} \sum_{p \in P: j \in L_p} u_{fp} \geq \sum_{j \in L_{\hat{p}}} u_{f\hat{p}} = |L_{\hat{p}}| u_{f\hat{p}}$, i.e., if an aircraft family f is assigned to a pairing \hat{p} , then all the $|L_{\hat{p}}|$ legs covered by this pairing must be flown by aircraft belonging to that family. Hence, (6.2.6) is a stronger representation than simply the latter restriction.

Constraint (6.2.7) ensures that each aircraft is assigned to at most one route, and Constraint (6.2.8) requires that the initial station of an aircraft and the departure station of the first leg on a route must be the same. Constraint (6.2.9) enforces that the number of aircraft of any type $a \in AT$ that start at a station $s \in S$ must equal the number of aircraft of this type that wrap around back to this station at the end of the planning horizon. Constraints (6.2.10) and (6.2.11) serve to define valid routes for each aircraft in the context of flow conservation, while Constraints (6.2.12) and (6.2.13) are the time-windows constraints that incorporate the flexibility of flight times in this model. Note that we tacitly assume here that these time-window variations for (re-)scheduling flights are sufficiently narrow and properly spaced so that the different defined duty periods remain valid with respect to mandated

work regulations. Finally, Constraint (6.2.14) imposes logical restrictions on the decision variables.

In addition, Model 4 can be extended to accommodate various work rules. For instance, in order to impose a specified minimum and maximum number of flying hours for crews on any given pairing p that is flown by aircraft of family f , we add the (logical) constraint:

$$\sum_{j \in L_p} \theta_{fj} \notin [\alpha_f, \beta_f] \quad \Rightarrow \quad u_{fp} = 0, \quad \forall p \in P, f = 1, \dots, \varphi. \quad (6.2.15)$$

Hence, (6.2.15) can be applied *a priori* fix certain u -variables to zero based on the relevant specified data.

Another possible consideration is to guarantee a minimum number of departures, ρ_{fs} , at a given crew base $s \in B$ for each fleet group f . By denoting δ_{ps} as the number of outbound legs at base s that are covered by pairing $p \in P$, we include the constraint:

$$\sum_{p \in P} \delta_{ps} u_{fp} \geq \rho_{fs}, \quad \forall s \in B, f = 1, \dots, \varphi. \quad (6.2.16)$$

This model can be enhanced using techniques similar to those discussed in Chapters 3-5, and suitable algorithms can likewise be designed. We relegate this study to future research.

Chapter 7

Conclusions

7.1 Summary and Closing Remarks

This dissertation has addressed the development and analysis of mathematical models that serve to integrate different combinations of the airline operational planning problems of schedule design, fleet assignment, aircraft routing, and crew scheduling, and the design and testing of effective solution methodologies. Three models have been investigated in detail for this purpose with computational results 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 fourth comprehensive integrated model has also been formulated, where we have relegated its analysis along similar lines for future research.

The first model studied herein integrates schedule design 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 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, which resulted in less than a 2% deterioration in solution quality, while reducing 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 design is performed separately before the fleet assignment stage by selecting a limited number 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 percentage 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).

The second model provided a further enhancement by integrating schedule design and fleet assignment, while incorporating additional features such as flight retiming, schedule balance, and demand recapture. This problem was formulated as a mixed-integer program and was enhanced by valid inequalities generated through a polyhedral analysis in concert with suitable separation routines, and a Benders decomposition solution approach was designed to facilitate the solution process. Using real data obtained from *United Airlines*, we have provided computational results to demonstrate the efficacy of the modeling and algorithmic strategies as well as the benefits of integrating each of the aforementioned features, both separately and combined. One of the benefits of flight retiming is to facilitate the generation of new connecting itineraries for serving certain markets, and thereby improve profits by judiciously assigning appropriate aircraft types to each of the individual flight legs. When flight times were permitted to shift by up to ± 10 minutes, the estimated increase in profit

was about \$14.9M/year over the baseline case where only original flight legs were used. Our experiments also revealed that the time-window discretizations need to be carefully selected to achieve better improvements, given a specified run-time limit. By placing leg copies 10 minutes apart, we were able to improve model tractability and thereby obtain solutions that were 1.02% better on average with respect to utilizing the same time-windows but at five minute intervals between the leg copies. To investigate such issues in practice, airlines need to consider in greater detail the variations in connection times and study the effect of connection times on demand. Moreover, when a new significant market emerges that needs to be explored, the time-window model can be used to appropriately revise a given schedule to accommodate this demand within the current operations, if profitable, and thereby increase aircraft utilization as well. Another important feature that can improve market shares is to provide schedule balance. Whereas this further restricts the model, it helps airlines preserve certain critical time-slots under FAA mandated “use-or-lose” rules, and moreover, the accompanying increase in demand can potentially improve profits. In our experiments, while considering two levels of such schedule balance restrictions, the results indicated a 1.52% and 0.49% increase in profits, respectively, over the baseline case. Whereas these results pertain to an optimistic 15% increase in demand on affected itineraries, a balanced schedule can improve customer goodwill and loyalty, and the consequent effect on forecasted demand is an issue that is worthy of further study. Next, we considered a detailed representation of demand recapture from a primary itinerary to other related itineraries, as opposed to simply approximating recapture as a fraction of the spill as done in some basic fleet assignment models. We also conducted a sensitivity analysis of the effect of recaptured demand with respect to the parameter δ that penalizes switches in itineraries. Using values of $\delta = 1, 50, 100,$ and 200 (dollars per passenger switched), our computational results demonstrated average respective proportions of 4.8%, 4.7%, 4.4%, and 4.1% in recaptured demand, thereby inducing additional profits of 2.10%, 2.09%, 2.02%, and 1.92%, respectively, over the baseline case. Moreover, this improvement in profits was accompanied by an increase in the average load factor of 6.7%, 6.5%, 5.6%, and 4.9%, respectively, due to a better matching of

demand with capacity. Re-accommodating passengers spilled from their originally desired itinerary schedules plays an important role in obtaining quality fleet assignment solutions, as well as provides a valuable tool to assess fleet acquisition plans of airlines in the long term. For further study, we recommended accommodating the consideration of demand recapture across fare-classes and between airlines, although this would entail more intricate data requirements. Overall, the results obtained from two intensity levels of the proposed SDFAM model that accommodate all the features of flight retiming, schedule balance, and demand recapture simultaneously, demonstrated a clear advantage by way of \$35.1 and \$31.8 million increases in annual profits, respectively, for the integrated models over the baseline case in which none of these additional features is considered. However, the largest size test instance for the higher intensity level case terminated with out-of-memory difficulties. This underscores the need for further research on designing more effective solution algorithms. In particular, Remark 4.1 in Section 4.4 discusses several ideas for accelerating the proposed Benders decomposition procedure that are worth investigating in this regard.

The third model adopted an alternative flight network modeling paradigm in lieu of the traditional time-space network to additionally integrate the design of aircraft routes, while incorporating various features such as itinerary-based demands, demand recapture, and through-flights. Maintenance routing considerations were also modeled in aggregate form to render more practical and feasible routing decisions. In order to enhance the model solvability, we introduced suitable hierarchical symmetry-breaking constraints as well as utilized different classes of valid inequalities including some generated to restructure the model using the Reformulation-Linearization Technique (RLT). A Benders decomposition-based methodology was developed to effectively solve the large-scale problem. We explored four different algorithmic variants, among which the best performing procedure (Algorithm A1) adopted two sequential levels of Benders partitioning method along with a special-ordered-set branching mechanism. We then applied Algorithm A1 to perform several extensive sensitivity analysis experiments to study the effects of different modeling features and algorithmic strategies. In one such experiment, we examined how the relative proportion of through-

flights impacts the overall net profit by assuming an increase in demand or the airfare that passengers are willing to pay for itineraries that contain through-flight leg pairs. The case $L_1^{TF}(A_{TF} = A_{TF}^1)$ that considers both mandatory and optional through-flight leg pairs in the model based on their relative effects on demands and enhanced revenues, achieved the best solution with an estimated increase in expected annual profits of \$2.4 million over the baseline case. In a second experiment, we tested the effect of utilizing symmetry-breaking constraints in concert with compatible objective perturbation terms that impose suitable hierarchical relationships among the aircraft of the same type. This composite strategy greatly enhanced problem solvability and improved the estimated annual profits by \$5.8 million over the baseline case. In a third experiment, we investigated the effect of demand recapture and its sensitivity to the corresponding cost parameter δ , to assess the benefits accruing from re-accommodating the demand spilled from primary itineraries to other compatible itineraries depending upon the fleet capacity and usage. The different values of $\delta = 100, 50, \text{ and } 1$ (dollars per re-routed passenger) induced average respective proportions of 3.2%, 3.4%, and 3.7% in recaptured demand, resulting in additional estimated annual profits of \$3.7 million, \$3.8 million, and \$4.0 million over the baseline case. Moreover, this improvement in profits reflected an increase in the average load factor of 2.7%, 3.2%, and 4.1%, respectively, thus displaying a better matching of capacity with demand. Finally, we performed an experiment to study the effect of incorporating the proposed valid inequalities within the model. The results revealed that the tighter model representation helped reduce the computational effort by 11% on average, while achieving better solutions that yielded on average an increase in estimated annual profits of \$1.4 million, thus exhibiting the utility of applying valid inequalities.

The fourth model considered a more comprehensive integration of the fleet assignment, aircraft routing, and crew scheduling processes. This integrated model can be easily augmented to accommodate various rules such as a minimum and maximum number of flying hours for crews on any given pairing, and a minimum number of departures at a given crew base for each fleet group. The analysis of this model and the design of suitable solution

algorithms are recommended for future research.

The results of this research have demonstrated significant potential impacts on profitability and service quality of airline companies by simultaneously considering issues related to schedule design, aircraft fleet assignment, and aircraft routing. By recognizing the interplay among these operational planning problems and integrating them within consolidated models (including the crew scheduling step), airlines will be able to make more profitable decisions over the traditional sequential process that examines these interrelated problems separately. Moreover, the solution methodologies developed for analyzing the formulated large-scale models also contribute toward the concepts and approaches for solving other related scheduling and discrete optimization problems.

Bibliography

1. Abara, J., (1989), “Applying Integer Programming to the Fleet Assignment Problem”, *Interfaces*, 19, 20-28.
2. Adams, W.P., and Sherali, H.D., (1990), “Linearization Strategies for a Class of Zero-one Mixed Integer Programming Problems”, *Operations Research*, 38(2), 217-226.
3. AhmadBeygi, S., Cohn, A., and Lapp, M., (2010), “Decreasing Airline Delay Propagation By Re-Allocating Scheduled Slack”, *IIE Transaction*, 42(7), 478-489.
4. Balas, E., (1974), “Disjunctive Programming: Properties of the Convex Hull of Feasible Points”, *Management Science Research Report*, 348, GSIA, Carnegie-Mellon University, Pittsburgh, PA.
5. Ball, P., Hoffman, K., and Rushmeir, R., (1996), “The Use of Column Generation in Solving Very Large Fleet Assignment Problems”, *INFORMS Conference*, Washington, D.C..
6. Barnhart, C., Boland, N.L., Clarke, L.W., Johnson, E.L., and Nemhauser, G.L., Sheno, R.G. (1998), “Flight String Models for Aircraft Fleeting and Routing”, *Transportation Science*, 32(3), 208-220.
7. Barnhart, C., Farahat, A., and Lohatepanont, M., (2009), “Airline Fleet Assignment with Enhanced Revenue Modeling”, *Operations Research*, 57(1), 231-244.

8. Barnhart, C., Kniker, T., and Lohatepanont, M., (2002b), "Itinerary-based Airline Fleet Assignment", *Transportation Science*, 36, 199-217.
9. Barnhart, C., Lu, F., and Shenoi, R., (1998c), "Integrated Airline Schedule Planning", In *Operations Research in the Airline Industry*, 384-403, Kluwer, Boston.
10. Barnhart, C., and Shenoi, R.G., (1998), "An Approximate Model and Solution Approach for the Long-haul Crew Pairing Problem", *Transportation Science*, 32(3), 221-231.
11. Benders, J.F., (1962), "Partitioning Procedures for Solving Mixed-variables Programming Problems" *Numerische Mathematik*, 4, 238-252.
12. Berge, M.E., and Hopperstad, C.A., (1993), "Demand Driven Dispatch: A Method for Dynamic Aircraft Capacity Assignment, Models and Algorithms", *Operations Research*, 41, 153-168.
13. Botimer, T.C., (1996), "Efficiency Considerations in Airline Pricing and Yield Management", *Transportation Research*, 30(4), 307-317.
14. Bratu, S., and Barnhart, C., (2006), "Flight Operations Recovery: New Approaches Considering Passenger Recovery", *Journal of Scheduling*, 9, 279-298.
15. Cao, J.-M., and Kanafani, A., (1997), "Real-time Decision Support for Integration of Airline Flight Cancellations and Delays, Parts I and II", *Transportation Planning and Technology*, 20, 183-201.
16. Chang, P.-C., Hsieh, J.-C., Yeh, C.-H., and Liu, C.-H., (2006) "A Case-Based Seat Allocation System for Airline Revenue Management", Huang, Li, and Irwin (Eds.): ICIC 2006, LNCS 4113, 1003-1011.
17. Clarke, L.W., Hane, C.A., Johnson, E.L., and Nemhauser, G.L., (1996), "Maintenance and Crew Considerations in Fleet Assignment", *Transportation Science*, 30, 249-260.

18. Clarke, L.W., Johnson, E.L., Nemhauser, G.L., and Zhu, Z., (1997), "The Aircraft Rotation Problem", *Annals of Operations Research, Mathematics of Industrial Systems II*, 69, 33-46.
19. Clarke, M., (1997), "An Introduction to the Airline Recovery Problem", Masters Thesis, Massachusetts Institute of Technology, Cambridge, MA.
20. Clausen, J., Larsen, A., Larsen, J., and Rezanova, N.J., (2010), "Disruption Management in the Airline Industry Concepts, Models and Methods", *Computers and Operations Research*, 37(5), 809-821.
21. Cohn, A.M., and Barnhart, C., (2003), "Improving Crew Scheduling by Incorporating Key Maintenance Routing Decisions", *Operations Research*, 51(3), 387-396.
22. Cordeau, J.F., Stojkovic, G., Soumis, F., and Desrosiers, J., (2001), "Benders Decomposition for Simultaneous Aircraft Routing and Crew Scheduling", *Transportation Science*, 35(4), 375-388.
23. Cote, G., and Laughton, M.A., (1984), "Large-scale Mixed Integer Programming: Benders-type Heuristics", *European Journal of Operational Research*, 16, 327-333.
24. Desaulniers, G., Desrosiers, J., Dumas, Y., Solomon, M., and Soumis, F., (1997b), "Daily Aircraft Routing and Scheduling", *Management Science*, 43, 841-855.
25. Du Merle, O., Villeneuve, D., Desrosiers, J., and Hansen, P., (1999), "Stabilized Column Generation", *Discrete Mathematics*, 194, 229-237.
26. Dunleavy, H., and Westermann, D., (2005), "Future of Revenue Management", *Journal of Revenue and Pricing Management*, 3(4), 380-383.
27. Erdmann, A., Nolte, A., Noltemeier, A. and Schrader, R., (2001), "Modeling and Solving an Airline Schedule Generation Problem", *Annals of Operations Research*, 107, 117-142.

28. Feo, T.A., and Bard, J.F., (1989), "Flight Scheduling and Maintenance Base Planning", *Management Science*, 35, 1414-1432.
29. Fischetti, M., and Lodi, A., (2003) "Local Branching", *Mathematical Programming*, 98(1-3), 23-47.
30. Fischetti, M., Salvagnin, D., and Zanette, A., (2010), "A Note on the Selection of Benders' Cuts", *Mathematical Programming B*, 124, 175-182.
31. Frank, M., Friedemann, M., Mederer, M., and Schroeder, A., (2006), "Airline Revenue Management: A Simulation of Dynamic Capacity Management", *Journal of Revenue and Pricing Management*, 5(1), 62-71.
32. Gamache, M., Hertz, A., and Ouellet, J.O., (2007), "A Graph Coloring Model for a Feasibility Problem in Monthly Crew Scheduling with Preferential Bidding", *Computers and Operations Research*, 34, 2384-2395.
33. Gamache, M., Soumis, F., Marquis, G., Desrosiers, J., (1999), "A Column Generation Approach for Large-scale Aircrew Rostering Problems", *Operations Research*, 47(2), 247-262.
34. Gao, C., (2007), "Airline Integrated Planning and Operations", Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, GA.
35. Geoffrion, A.M., and Graves, G.W., (1974), "Multicommodity Distribution System Design by Benders Decomposition", *Management Science*, 20(5), 822-844.
36. Ghoniem, A., and Sherali, H.D., (2009) "Defeating Symmetry Via Objective Perturbations and Hierarchical Constraints", Manuscript, Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA.
37. Gopalan, R., and Talluri, K.T., (1998a), "Mathematical Models in Airline Schedule Planning: A Survey", *Annals of Operations Research*, 76, 155-185.

38. Gopalan, R., and Talluri, K.T., (1998b), “The Aircraft Maintenance Routing Problem”, *Operations Research*, 46, 260-271.
39. Gosavi, A., Bandla, N., and Das, T.K., (2002), “A Reinforcement Learning Approach to a Single Leg Airline Revenue Management Problem with Multiple Fare Classes and Overbooking”, *IIE Transaction*, 34, 729-742.
40. Hane, C.A., Barnhart, C., Johnson, E.L., Marsten, R.E., Nemhauser, G.L., and Sigismondi, G., (1995), “The Fleet Assignment Problem: Solving a Large-scale Integer Program”, *Mathematical Programming*, 70, 211-232.
41. Haouari, M., Sherali, H.D., Mansour, F.Z., and Aissaoui, N., (2010), “Exact Approaches for Integrated Aircraft Fleeting and Routing”, *Computational Optimization and Applications* DOI 10.1007/s10589-009-9292-z.
42. Jacob, T., Johnson, E., and Smith, B., (1999) O&D FAM: Incorporating Passenger Flow into the Fleeting Process. *Thirty-Ninth Annual AGIFORS Sympos.*, New Orleans, LA.
43. Jarrah, A., Yu, G., Krishnamurthy, N., and Rakshit, A., (1993), “A Decision Support Framework for Airline Flight Cancellations and Delays”, *Transportation Science*, 27, 266-280.
44. Jiang, H., and Barnhart, C., (2009), “Dynamic Airline Scheduling”, *Transportation Science*, 43(3), 336-354.
45. Kang, L.S., (2004), “Degradable Airline Scheduling: an Approach to Improve Operational Robustness and Differentiate Service Quality”, Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.
46. Kim, D., and Barnhart, C. (2007), “Flight Schedule Design for a Charter Airline”, *Computers and Operations Research*, 34, 1516-1531.

47. Klabjan, D., Johnson, E.L., and Nemhauser, G.L., (2002), "Airline Crew Scheduling with Time Windows and Plane Count Constraints", *Transportation Science*, 36, 337-348.
48. Kniker, T., (1998), "Itinerary-based Airline Fleet Assignment", Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.
49. Lan, S., Clarke, J.-P., and Barnhart, C., (2006), "Planning for Robust Airline Operations: Optimizing Aircraft Routing and Flight Departure Times to Minimize Passenger Disruptions", *Transportation Science*, 40(1), 15-28.
50. Lettovsky L., (1997), "Airline Operations Recovery: an Optimization Approach", Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, GA.
51. Levin, A., (1971), "Scheduling and Fleet Routing Models for Transportation Systems", *Transportation Science*, 5, 233-255.
52. Lohatepanont, M., and Barnhart, C., (2004), "Airline Schedule Planning: Integrated Models and Algorithms for Schedule Design and Fleet Assignment", *Transportation Science*, 38, 19-32.
53. Magnanti, T.L., and Wong, R.T. (1981), "Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria", *Operations Research*, 29, 464-484.
54. McDaniel, D., and Devine, M. (1977), "A Modified Benders Partitioning Algorithm for Mixed Integer Programming", *Management Science*, 24(3), 312-319.
55. Mercier, A., Cordeau, J.F., and Soumis, F., (2005), "A Computational Study of Benders Decomposition for the Integrated Aircraft Routing and Crew Scheduling Problem", *Computers and Operations Research*, 32, 1451-1476.
56. Mercier, A., and Soumis, F. (2007), "An Integrated Aircraft Routing, Crew Scheduling and Flight Retiming Model", *Computers and Operations Research*, 34, 2251-2265.

57. Nissen, R., and Haase, K., (2006), "Duty-period-based Network Model for Crew Rescheduling in European Airlines", *Journal of Scheduling*, 9, 255-278.
58. Papadakos, N., (2009), "Integrated Airline Scheduling", *Computers and Operations Research*, 36(1), 176-195.
59. Parker, G. (2003), "Optimising Airline Revenue Management", *Journal of Revenue and Pricing Management*, 2(2), 138-149.
60. Pilla, V.L., Rosenberger, J.M., Chen, V.C.P., and Smith, B., (2008), "A Statistical Computer Experiments Approach to Airline Fleet Assignment", *IIE Transaction*, 40, 524-537.
61. Pulugurtha, S.S., and Nambisan, S.S., (2003), "A Decision-Support Tool for Airline Yield Management Using Genetic Algorithms", *Computer-Aided Civil and Infrastructure Engineering*, 18, 214-223.
62. Rei, W., Cordeau, J.-F., Gendreau, M., and Soriano, P., (2009) "Accelerating Benders Decomposition by Local Branching", *INFORMS Journal on Computing*, 21(2), 333-345.
63. Rexing, B., Barnhart, C., Kniker, T.S., Jarrah, T.A., and Krishnamurthy, N., (2000), "Airline Fleet Assignment with Time Windows", *Transportation Science*, 34, 1-20.
64. Rushmeier, R.A., and Kontogiorgis, S.A., (1997), "Advances in the Optimization of Airline Fleet Assignment", *Transportation Science*, 31, 159-169.
65. Saharidis, G.K.D., and Ierapetritou, M.G., (2010), "Improving Benders Decomposition Using Maximum Feasible Subsystem (MFS) Cut Generation Strategy", *Computers and Chemical Engineering*, 34, 1237-1245.
66. Saharidis, G.K.D., Minoux, M., and Ierapetritou, M.G., (2010), "Accelerating Benders Method Using Covering Cut Bundle Generation", *International Transactions in Operational Research*, 17(2), 221-237.

67. Sandhu, R., and Klabjan, D., (2007), "Integrated Airline Fleeting and Crew-Pairing Decisions", *Operations Research*, 55, 439-456.
68. Schwind, M., and Wendt, O., (2002), "Dynamic Pricing of Information Products Based on Reinforcement Learning: A Yield-Management Approach", M. Jarke et al. (Eds.): KI 2002, LNAI 2479, 5166.
69. Sen, S., and Sherali, H.D., (2006), "Decomposition with Branch-and-Cut Approaches for Two Stage Stochastic Mixed-integer Programming", *Mathematical Programming*, 106(2), 203-223.
70. Sherali, H.D., (1982), "Equivalent Weights for Lexicographic Multiple Objective Programs: Characterizations and Computations", *European Journal of Operational Research*, 11(4), 367-379.
71. Sherali, H.D., and Adams, W.P., (1990), "A Hierarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems", *SIAM Journal on Discrete Mathematics*, 3(3), 411-430.
72. Sherali, H.D., and Adams, W.P., (1994), "A Hierarchy of Relaxations and Convex Hull Characterizations for Mixed-Integer Zero-One Programming Problems", *Discrete Applied Mathematics*, 52, 83-106.
73. Sherali, H.D., and Adams, W.P., (1999), "Reformulation-Linearization Techniques for Discrete Optimization Problems", *Handbook of Combinatorial Optimization*, eds. D.-Z. Du and P. M. Pardalos, Kluwer Academic Publishers, Vol. 1, pp. 479-532.
74. Sherali, H.D., Bae, K.-H., and Haouari, M., (2009), "Integrated Airline Schedule Design and Fleet Assignment: Polyhedral Analysis and Benders Decomposition Approach", *INFORMS Journal on Computing*, forthcoming.
75. Sherali, H.D., Bae, K.-H., and Haouari, M., (2010a), "A Benders Decomposition Approach for an Integrated Airline Schedule Design and Fleet Assignment Problem with

- Flight Retiming, Schedule Balance, and Demand Recapture”, Submitted to *Annals of Operations Research*.
76. Sherali, H.D., Bae, K.-H., and Haouari, M., (2010b), “An Integrated Approach for Airline Schedule Design, Fleet Assignment, and Aircraft Routing”, Manuscript, Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA.
 77. Sherali, H.D., Bish, E.K., and Zhu, X., (2005), “Polyhedral Analysis and Algorithms for a Demand Driven Refueling Model for Aircraft Assignment”, *Transportation Science*, 39(3), 349-366.
 78. Sherali, H.D., Bish, E.K., and Zhu, X., (2006) “Airline Fleet Assignment Concepts, Models, and Algorithms”, *European Journal of Operational Research*, 172(1), 1-30.
 79. Sherali, H.D., and Lunday, B., (2010), “On Generating Maximal Nondominated Benders Cuts”, Manuscript, Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA.
 80. Sherali, H.D., and Shetty, C.M. (1980), “*Optimization with Disjunctive Constraints*”, Lecture Notes in *Economics and Mathematical Systems*, 181, Springer Verlag, Berlin-Heidelberg-New York.
 81. Sherali, H.D. and Smith, J.C. (2001), “Improving Discrete Model Representations Via Symmetry Considerations”, *Management Science*, 47, 1396-1407.
 82. Sherali, H.D., and Soyster A.L., (1983), “Preemptive and Nonpreemptive Multi-Objective Programs: Relationships and Counter Examples”, *Journal of Optimization Theory and Applications*, 39(2), 173-186.
 83. Sherali, H.D., and Zhu, X. (2008), “Two-Stage Fleet Assignment Model Considering Stochastic Passenger Demands”, *Operations Research*, 56(2), 383-399.

84. Smith, B.C., Leimkuhler, J.F., and Darrow, R.M., (1992), "Yield Management at American Airlines", *Interfaces*, 22(1), 8-31.
85. Stojkovic, M., Soumis, F., and Desrosiers, J., (1998), "The Operational Airline Crew Scheduling Problem", *Transportation Science*, 32, 232-245.
86. Subramanian, R., Scheff Jr, R.P., Quillinan, J.D., Wiper, D.S., and Marsten, R.E., (1994), "Coldstart: Fleet Assignment at Delta Airlines", *Interfaces*, 24(1), 104-120.
87. Subramanian, S., and Sherali, H.D., (2008), "An Effective Deflected Subgradient Optimization Scheme for Implementing Column Generation for Large-Scale Airline Crew Scheduling Problems", *INFORMS Journal on Computing*. 20(4), 565-578.
88. Talluri, K., (1996), "Swapping Applications in a Daily Airline Fleet Assignment", *Transportation Science*, 30, 237-248.
89. Talluri, K., (1998), "The Four-Day Aircraft Maintenance Routing Problem", *Transportation Science*, 32(1), 43-53.
90. Teodorovic, D., and Krcmar-Nozic, E., (1989), "Multicriteria Model to Determine Flight Frequencies on an Airline Network under Competitive Conditions", *Transportation Science*, 23(1), 14-25.
91. Teodorovic, D., and Stojkovic, G., (1995), "Model to Reduce Airline Schedule Disturbances", *Journal of Transportation Engineering*, 121, 324331.
92. Thengvall, B., Bard, J., and Yu, G., (2000), "Balancing User Preferences for Aircraft Schedule Recovery During Irregular Operations", *IIE Transactions*, 32, 181193.
93. Vance, P.H., Barnhart, C., Johnson, E.L., Nemhauser, G.L., (1997), "Airline Crew Scheduling: A New Formulation and Decomposition Algorithm", *Operations Research*, 45(2), 188-200.

94. Weatherford, L.R., and Belobaba, P.P., (2002), "Revenue Impacts of Fare Input and Demand Forecast Accuracy in Airline Yield Management", *Journal of the Operational Research Society*, 53, 811-821.
95. Weatherford, L.R., and Bodily, S.E., (1992), "A Taxonomy and Research Overview of Perishable-asset Revenue Management: Yield Management, Overbooking, and Pricing", *Operations Research*, 40(5), 831-844.
96. Yan, S., and Young, H.F., (1996), "A Decision Support Framework for Multi-fleet Routing and Multi-stop Flight Scheduling", *Transportation Research*, 30A, 379-398.
97. Zaki, H., (2000), "Forecasting for Airline Revenue Management", *The Journal of Business Forecasting Methods & Systems*, 19(1), 2-6.