Integrated Aircraft Fleeting, Routing, and Crew Pairing Models and Algorithms for the Airline Industry

Shengzhi Shao

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
Kimberly P. Ellis
Barbara M. P. Fraticelli
Antonio A. Trani

December 6th, 2012
Blacksburg, Virginia

Keywords: Airline operations research, integrated airline scheduling, compact formulation, fleet assignment, aircraft routing, crew pairing, itinerary-based passenger mix, Reformulation-Linearization Technique (RLT), Benders decomposition, subgradient optimization, branch-and-price, large-scale optimization.

Integrated Aircraft Fleeting, Routing, and Crew Pairing Models and Algorithms for the Airline Industry

Shengzhi Shao

(ABSTRACT)

The air transportation market has been growing steadily for the past three decades since the airline deregulation in 1978. With competition also becoming more intense, airline companies have been trying to enhance their market shares and profit margins by composing favorable flight schedules and by efficiently allocating their resources of aircraft and crews so as to reduce operational costs. In practice, this is achieved based on demand forecasts and resource availabilities through a structured *airline scheduling process* that is comprised of four decision stages: schedule planning, fleet assignment, aircraft routing, and crew scheduling. The outputs of this process are flight schedules along with associated assignments of aircraft and crews that maximize the total expected profit.

Traditionally, airlines deal with these four operational scheduling stages in a sequential manner. However, there exist obvious interdependencies among these stages so that restrictive solutions from preceding stages are likely to limit the scope of decisions for succeeding stages, thus leading to suboptimal results and even infeasibilities. To overcome this drawback, we first study the aircraft routing problem, and develop some novel modeling foundations based on which we construct and analyze an integrated model that incorporates fleet assignment, aircraft routing, and crew pairing within a single framework.

Given a set of flights to be covered by a specific fleet type, the *aircraft routing problem* (ARP) determines a flight sequence for each individual aircraft in this fleet, while incorporating specific considerations of minimum turn-time and maintenance checks, as well as restrictions on the total accumulated flying time, the total number of takeoffs, and the total number of days between two consecutive maintenance operations. This stage is significant to airline companies as it directly assigns routes and maintenance breaks for each aircraft in service. Most approaches for solving this problem adopt set partitioning formulations that include exponentially many variables, thus requiring the design of specialized column generation or branch-and-price algorithms. In this dissertation, however, we present a novel compact polynomially sized representation for the ARP, which is then linearized and lifted using the Reformulation-Linearization Technique (RLT). The resulting formulation remains polynomial in size, and we show that it can be solved very efficiently by commercial software without complicated algorithmic implementations. Our numerical experiments using real data obtained from *United Airlines* demonstrate significant savings in computational effort; for example, for a daily network involving 344 flights, our approach required only about 10 CPU seconds for deriving an optimal solution.

We next extend Model ARP to incorporate its preceding and succeeding decision stages, i.e., fleet assignment and crew pairing, within an integrated framework. We formulate a suitable representation for the integrated fleeting, routing, and crew pairing problem (FRC), which accommodates a set of fleet types in a compact manner similar to that used for constructing the aforementioned aircraft routing model, and we generate eligible crew pairings on-the-fly within a set partitioning framework. Furthermore, to better represent industrial practice, we incorporate itinerary-based passenger demands for different fare-classes. The large size of the resulting model obviates a direct solution using off-the-shelf software; hence, we design a solution approach based on Benders decomposition and column generation using several acceleration techniques along with a branch-and-price heuristic for effectively deriving a solution to this model. In order to demonstrate the efficacy of the proposed model and solution approach and to provide insights for the airline industry, we generated several test instances using historical data obtained from *United Airlines*. Computational results reveal that the massively-sized integrated model can be effectively solved in reasonable times ranging from several minutes to about ten hours, depending on the size and structure of the instance. Moreover, our benchmark results demonstrate an average of 2.73% improvement in total profit (which translates to about 43 million dollars per year) over a partially integrated approach that combines the fleeting and routing decisions, but solves the crew pairing problem sequentially. This improvement is observed to accrue due to the fact that the fully integrated model effectively explores alternative fleet assignment decisions that better utilize available resources and yield significantly lower crew costs.

Keywords: Airline operations research, integrated airline scheduling, compact formulation, fleet assignment, aircraft routing, crew pairing, itinerary-based passenger-mix, Reformulation-Linearization Technique (RLT), Benders decomposition, subgradient optimization, branch-and-price, large-scale optimization.

Dedication

To my family, for their continual support from the very beginning.

Acknowledgements

First and foremost I would like to express my deepest gratitude toward my advisor, Dr. Hanif D. Sherali, an outstanding researcher, mentor, and gentleman. It has been an honor to be his Ph.D. student and to work with him. He has taught me, consciously and unconsciously, how to perform good research, and many other things beyond academics. I would never have been able to complete this work without his guidance and inspiration.

Moreover, I am also grateful to my committee members for their help throughout these years. In specific, I thank Dr. Kimberly P. Ellis for her support and kind-heartedness–it was a pleasure working with her. Thanks also to Dr. Barbara M. P. Fraticelli for her excellent teaching skills and helpful discussions on the broad scope of optimization. Furthermore, I appreciate Dr. Antonio A. Trani for his interesting lectures that walked me through topics within the realm of aviation research. I am indebted to all of them for their time, interest, and valuable insights on this work. Additionally, I sincerely acknowledge Dr. Mohamed Haouari for his insightful comments and suggestions toward this work. Also, many thanks to the *National Science Foundation* for supporting this research under Grant Number CMMI-0754236.

In addition, I am also thankful to several colleagues at Virginia Tech. I feel fortunate to have as my friends Lingrui Liao, Jason Judd, and many other peer students who suffered from my frequent interruptions but patiently helped me resolve difficulties. I also gratefully acknowledge my friend Xiang Ma in Nanjing, China, for his timely help with regard to computer programming issues. In addition, I appreciate the company of my other friends and the Chinese community in Blacksburg, who made my life here enjoyable and memorable.

Finally but most importantly, I would like to thank my family for all their love, patience, and encouragement. I thank my wife for faithfully standing with me throughout the past six years, sharing each and every joyful and not-that-joyful moment. And I lack words with which to express my gratitude toward my parents, who have raised me with all they have, and always been my biggest support. To them, I would like to dedicate this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Industrial Environment

The airline industry has been expanding steadily for the past thirty years, benefiting from demand spurred by a growing economy along with technological advances that have improved the overall flying experience. Although the increasing trend in demand was temporarily interrupted by the 9/11 event in 2001, the statistical data displayed in Fig. 1.1 reveals that the annual gross output of the air transportation industry had doubled over the two decades prior to the economic recession of 2008 (Bureau of Economic Analysis, 2011), while the demand has steadily increased.

From another perspective, however, since the deregulation in 1978, airline companies have been facing fiercer competition from their peers as well as from newly-emerging low-cost carriers (i.e., budget airlines). Some airlines have lost their market shares due to direct competition on busy origin-destination routes, which has also drastically reduced fare prices to levels even lower than marginal costs. In addition, the operational costs have been rising. As indicated in Fig. 1.2 (Bureau of Transportation Statistics, 2011), the average annual wage for all the 58,000 pilots, copilots, and flight engineers has doubled since 1990, and has reached $117,000 in 2010 (Bureau of Labor Statistics, 2011). Meanwhile, the fuel cost in recent years has also increased significantly due to the spike in the international crude oil index. Moreover, the most recent decrease in demand, as displayed in Fig. 1.3 (Bureau of Transportation

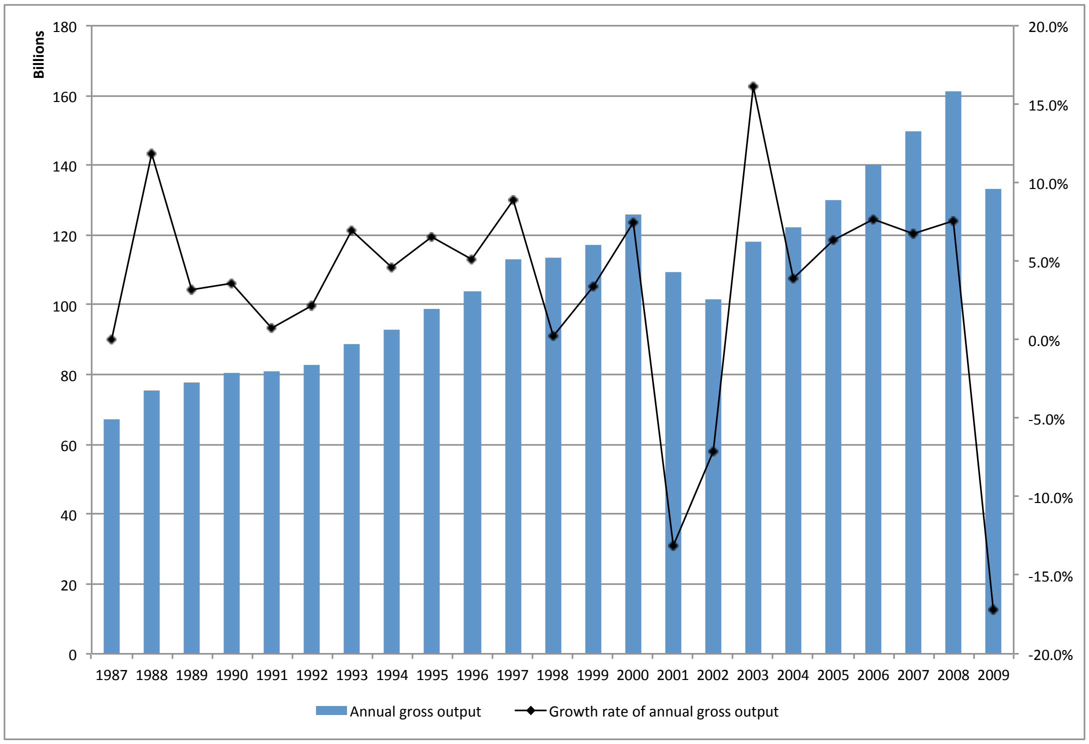

Figure 1.1: Air Transportation Annual Gross Output (1988-2009).

Statistics, 2011), has also contributed toward reducing the profit margin of the entire industry as a whole. Therefore, besides exploring merger and acquisition opportunities within the industry, airline companies have been actively striving to reduce operational costs in order to enhance profit margins.

Airline operations involve a diversity of aircraft of different types along with certified cockpit and cabin crews. An aircraft fleet (or simply *fleet*, or *aircraft type*) is comprised of identical aircraft having the same operational cost, cockpit configurations, and seating capacity; therefore, these aircraft can be assigned interchangeably and be maintained following the same maintenance regulations. Moreover, an *aircraft family* refers to a set of fleets that share the same cockpit configurations while having varying features such as the seating capacity. Usually, a crew member is only certified to operate aircraft within a specific family. For example, the Boeing 737 family contains aircraft types including 737-300, 737-500, 737-700, and 737-900ER, which, despite the differences in size and range, can be operated by any crew eligible to operate the 737 family.

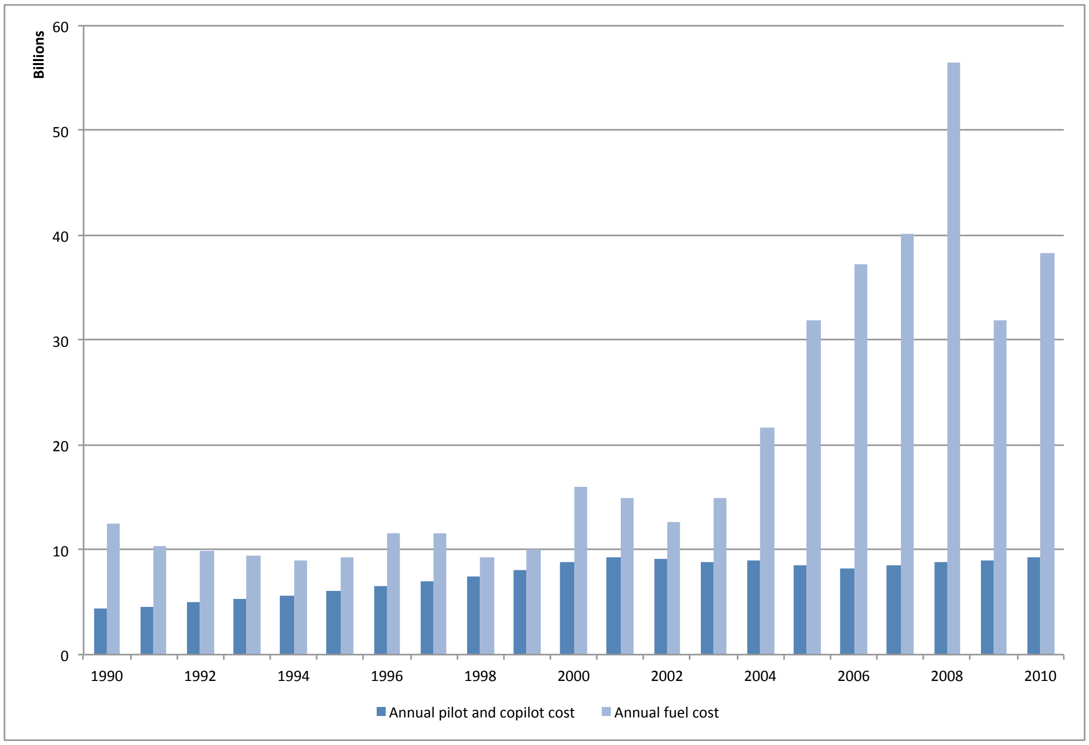

Figure 1.2: Annual Fuel and Pilot Costs for Large Certified US Air Carriers (1990-2011).

Large airline companies in North America typically run a hub-and-spoke type flight network, where a few major airports at large cities serve as hubs and other destinations as the spokes. Hubs are connected by dense flights using wide-body aircraft such as the Boeing 777 and the Airbus 330 in order to accommodate large travel demands between major cities, while passengers from surrounding areas are first transported to some nearby hub via narrow-body aircraft or regional jets before connecting to outbound flights toward their final destinations. Each flight in the network must be served by an appropriate aircraft and a corresponding crew in order to satisfy the travel demand to the extent possible. The associated incurred fuel and crew costs account for two significant components of the total operational cost for an airline, as demonstrated in Fig. 1.4. Hence, the incentive for reducing these operational costs drives airline companies to actively seek effective aircraft and crew assignments through the airline scheduling process.

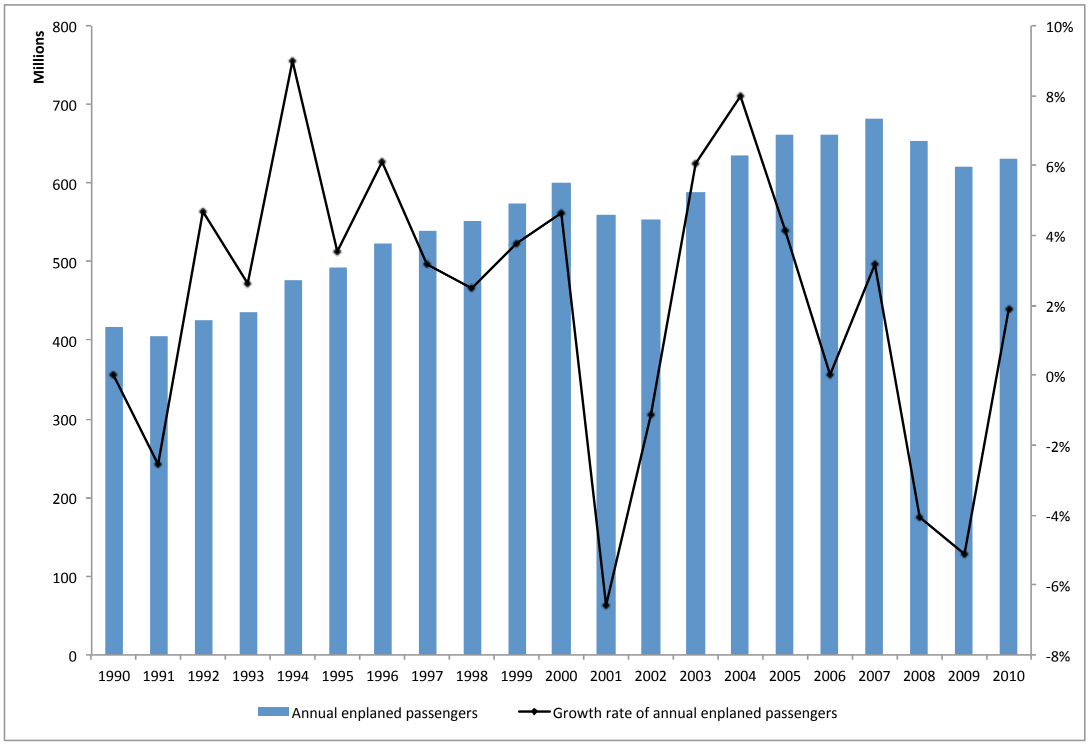

Figure 1.3: Annual Number of Enplaned Passengers in the Domestic Market (1990-2010).

1.2. Airline Scheduling Process and Research Motivation

As displayed in Fig. 1.5, the *airline scheduling process*, with the objective of maximizing the total profit (or minimizing the total cost), is often decomposed into four separate stages (Yu, 1998). First, nine to twelve months prior to the actual departure, a flight schedule is designed during the *schedule planning* stage based on strategic market competition forecasts. Next, in the *fleet assignment* stage, each fleet type is assigned to flight legs corresponding to the given flight schedule according to the aircraft's capacity and operating cost in order to maximize the total ticket revenue minus the cost. The output of this stage partitions the entire flight network into sub-networks according to each fleet type. Subsequently, within each sub-network, a *maintenance-feasible flight rotation* (or a sequence of flights) for each individual aircraft of the corresponding type is determined in the *aircraft routing* stage based on a series of maintenance requirements as mandated by the *Federal Aviation Administration* (FAA).

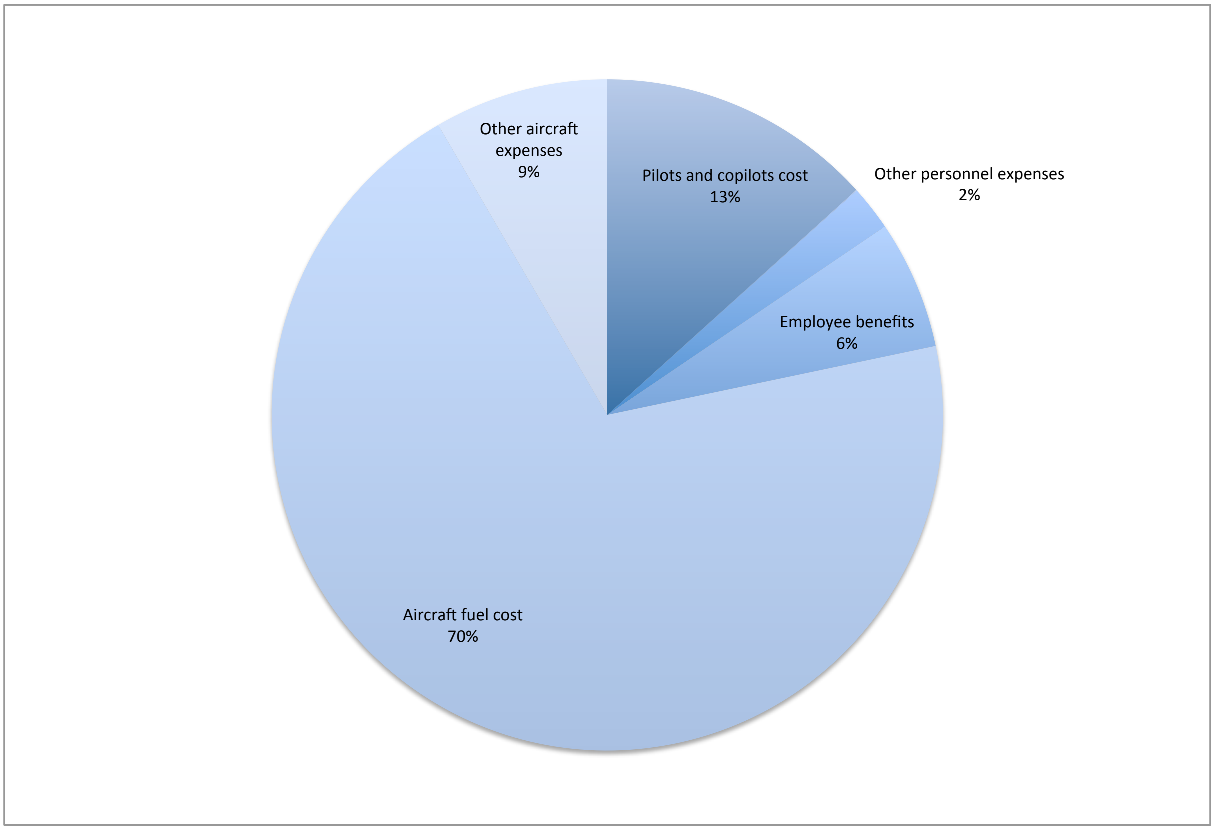

Figure 1.4: Principal Components of Operational Costs for Major US Airlines (2011).

Finally, in the *crew scheduling* stage, which is conducted for each aircraft family, crews are assigned to feasible pairings, where each *pairing* is comprised of *flight duties* (a sequence of flights served over a day's work) all interspersed with adequate rest periods. Such pairings are generated subject to several complex FAA-mandated regulations and work-rules, and optimized with respect to a nonlinear union-driven cost structure.

Although the foregoing operational phases are *implemented* sequentially in practice, their interdependences would naturally lead a purely sequential *decision-making* approach to suboptimal solutions, because prefixed decisions that are made while ignoring downstream considerations would tend to suboptimally restrict the ensuing decision stages. This might even result in infeasibility at some subsequent stages in the process. From this perspective, it is prudent to investigate models that integrate the different stages (or suitable combinations thereof) within a single framework in order to obtain improved solutions, while being cognizant of the fact that the problem complexity will also substantially increase as more aspects and decisions are considered simultaneously.

We also note that this airline scheduling process is heavily influenced by customer demands, which are random variables and are estimated by a heterogeneous process called *revenue management* (Jacobs et al., 2008). Moreover, the capacity of the assigned aircraft affects the total number of seats reserved for each fare-class, therefore potentially limiting the flight profitability. Furthermore, in practice, a common approach of *demand-driven dispatch* (D^3) is performed close to the departure day, in which aircraft within the same fleet are swapped according to realized booking records in order to accommodate additional travel demand and hence achieve increased revenue levels (Shebalov, 2009). Observe that the possibility of such swaps is highly dependent on the fleet assignment and routing decisions.

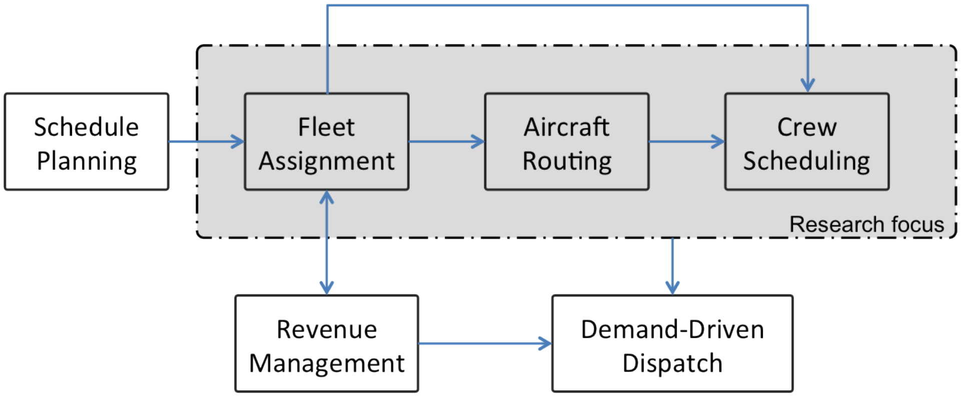

Figure 1.5: The Airline Scheduling Process

With the overall motivation of designing mechanisms for deriving improved solutions to the airline scheduling process, we first investigate the *aircraft routing problem*, which serves as the foundation of this dissertation. We propose a novel mixed-integer linear program (MIP) that is polynomially-sized and can be conveniently and effectively solved using off-the-shelf commercial MIP solvers such as CPLEX. Subsequently, we integrate this model with its preceding decision-making stage of fleet assignment and its succeeding stage of crew pairing in order to accommodate the inherent interactions among these three stages. This integrated approach provides improved, implementable solutions, in comparison with the myopic solutions that result from a sequential decision-making process, hence potentially enhancing the profitability of airline companies.

1.3. Dissertation Structure

The structure of this dissertation is organized as follows. In Chapter 2, we survey the existing literature on research pertaining to the airline scheduling process. In particular, we focus on models related to the aircraft routing problem, along with partially integrated formulations involving it, as well as on fully integrated models that include fleeting, routing, and crew pairing decisions within a single framework. Next, in Chapter 3, we develop a compact formulation for the aircraft routing problem that accommodates several FAA-mandated maintenance requirements, and we reformulate this model using suitable transformations along with the generation of classes of valid inequalities to enhance its solvability. Chapter 4 then integrates the aforementioned aircraft routing model with fleet assignment and crew pairing considerations, where we also include itinerary-based demands for better modeling the demand structure. Suitable model tightening approaches are introduced together with state-of-the-art solution techniques in order to effectively deal with the resulting large-scale formulation, and benchmark results using real data obtained from a major US airline company (*United Airlines*) are presented to demonstrate the effectiveness of the proposed approach. Finally, in Chapter 5, we summarize our research contributions along with conclusions and recommendations for future research directions in this fertile domain.

Chapter 2

Literature Review

In this chapter, we survey the existing literature on the airline scheduling process along with developed solution techniques, while focusing on the aircraft routing problem and related integrated models.

2.1. Aircraft Routing Problem

The airline scheduling problem plays a crucial role in the airline industry since a major portion of the profits generated in this industry can be attributed to effective and robust planning. Usually, the scheduling process begins 12 months in advance of operations, and the final schedule for each individual aircraft and crew are not fixed until a few weeks prior to implementation. Moreover, due to the rapid growth of the airline industry as well as due to the inherent complexity of the airline scheduling problem, even a moderate-size airline scheduling model is unlikely to be tractable by standard, direct solution methodologies. Therefore, the entire decision-making process is decomposed into four independent stages that are frequently solved sequentially, i.e., schedule planning, fleet assignment, aircraft routing, and crew scheduling (Yu, 1998). The outcomes of the process include a flight timetable and an assignment of aircraft and crews that cover each of the specified flights, while satisfying respective requirements. Section 2.2 presents a comprehensive review of the airline scheduling problem as described in detail by Klabjan (2005).

2.1.1 Problem Definition

Given that each flight leg has been assigned a specific type of aircraft, the *aircraft routing problem* (ARP) seeks to determine the flight sequence of each individual aircraft, or *tail number*, over a certain study period, in order to serve a specified set of flight legs while satisfying various maintenance requirements mandated by FAA. There are four types of such required maintenance checks. The type A check involves a routine visual inspection of major systems and is conducted every 65 flight hours, while the type B check is performed every 300 to 600 flight hours, and includes a complete visual inspection and a thorough lubrication of all moving parts. (We note that in the industry, these two routine checks are sometimes referred to as the *three-service check* and the A-check, respectively; however, in the following content, we will continue to refer to them as type A and B checks.) On the other hand, type C and type D checks take weeks to perform and involve a set of more rigorous procedures, the timings of which are usually planned at a higher level and are thus not included in the consideration of daily operations (Feo and Bard, 1989).

The main objective of the ARP is to minimize the total cost of assigning tail numbers to routes and performing maintenance checks. Tail number assignments incur a *through-value* benefit plus a penalty for undesired connections. The through-value can be viewed as a negative cost that accrues when connecting passengers do not need to change planes at intermediate stations. Staying on the same aircraft is appealing to passengers and thus generates extra revenues (Clarke et al., 1997). In addition, a penalty cost is incurred for *restricted flight connections*, the duration of which is only slightly more than the minimum turn-time of the particular aircraft, since tight connections can potentially cause delays and disruptions in operations, and can lead to infeasibility in the subsequent decision stage of crew pairing (Klabjan et al., 2002; Cohn and Barnhart, 2003; Mercier et al., 2005). Finally, a maintenance cost is incurred for each maintenance activity, and might also depend on whether the performed maintenance is either too much prior to, or too close to, the maximal flying hours, because the former wastes resources and labor man-hours due to excessively frequent maintenance operations, and the latter induces inflexibility and a lack of robustness in routing. It is worth noting that the objective function used for the ARP varies according to the emphasis of the model, and does not necessarily include all the costs listed above. In fact, the ARP is often posed simply as a feasibility problem (e.g., see Gopalan and Talluri

(1998); Talluri (1998)), which is the principal approach adopted herein in order to focus on the aspect of determining maintenance-feasible routes for aircraft (however, we also present computational results in Section 3.4 for using a suitable objective function in concert with our proposed model).

2.1.2 Existing ARP Models and Solution Techniques

The ARP is usually modeled within the framework of a time-space network in which each station is represented by a time-line spanning the study period. A node in the time-space network denotes an event of departure or arrival of a flight at a station at a specific time, and a directed arc between two stations represents a flight leg. An assignment cost is associated with each connection arc that represents a connection between a pair of flights at the same station, and a maintenance cost is assigned to each overnight ground arc that permits a maintenance activity. Typically, the ARP in the literature is formulated as a multicommodity network flow problem or a set partitioning problem (Klabjan, 2005). The multicommodity flow network has a polynomial space complexity, and is therefore more tractable by commercial solvers. On the other hand, the set partitioning formulation yields a tighter representation but involves an implicit enumeration of exponentially many routes and thus requires more complex solution techniques such as branch-and-price (Barnhart et al., 1998b). However, this latter model is more amenable to incorporating additional features of related interest.

Feo and Bard (1989) first proposed a model that combines routing decisions with maintenance base locations in both a finite and an infinite time horizon. Their model minimized the sum of various maintenance costs for each aircraft plus the fixed cost of opening maintenance bases. In this model, aircraft were assigned to each OD pair, i.e., a sequence of flight legs delineated by an initial and terminal flight within a day, without consideration of intermediate connections. The authors proposed a heuristic method, whereby thousands of eligible routes were randomly generated, and promising OD pairs were selected by solving a set covering problem. Focusing on a hub-and-spoke system, Daskin and Panayotopoulos (1989) specified a route as a sequence of flight legs that originates from and ends at the same hub station, to which each tail number was directly assigned. In this model, not all the flights were necessarily required to be covered since deleting some unattractive ones might increase profits. Furthermore, Clarke et al. (1997) incorporated through-values within their objective function. In order to maintain a balanced utilization of aircraft, their model required a continuous rotation that covers all the flight legs, i.e., an Eulerian tour for each fleet type, which led to solving an asymmetric traveling salesman problem over the connection network. The authors proposed node aggregation and arc aggregation during the preprocessing phase, and employed Lagrangian relaxation to solve the model. Desaulniers et al. (1997) accommodated variable departure time-windows within the ARP. Assuming that maintenance can be performed at particular stations during the night, they respectively formulated a node-arc and a node-path model with the objective of maximizing the total anticipated profit for aircraft assignment in a daily schedule context. The two formulations were proved to be equivalent, where the latter can be derived from the former by column generation. Furthermore, Ioachim et al. (1999) extended the study period for fleet routing to a week while considering schedule synchronization. They formulated this problem as a classical multiple traveling salesman model with time-windows, but did not incorporate mandatory maintenance requirements. Yan (2002) addressed fleet routing decisions within a layered passenger demand network, in which possible flight legs were selected for service based on capacity, demand, and profit considerations. Elf et al. (2003) studied aircraft rotation planning for European flight networks, and proposed a model for minimizing the total delay risk while restricting the frequency of visits to critical airports that handle intensive air traffic, and also while accommodating the possibility of delay accumulation along a flight path. Sriram and Haghani (2003) developed a model that accounts for both type A and type B maintenance checks. Similar to the model in Feo and Bard (1989), their model used a series of OD pairs within a multi-day multicommodity network flow framework. Possible aircraft re-assignments were also considered by ascribing penalties to each arc. Sarac et al. (2006) addressed an operational maintenance routing problem within a day for only *high-time aircraft*, i.e., aircraft that have accumulated flying hours up to a predefined threshold. The objective of the model was to optimally utilize the legal flying time while satisfying type A and type B maintenance checks. Another perspective was provided by Lan et al. (2006). Instead of optimizing the through-values, which are hard to capture, the authors focused on minimizing the total passenger disruptions by intelligently routing aircraft and selecting flight departure times. Their robust aircraft maintenance routing model attempted to absorb the propagated delay by the slack in connection times obtained using information from historical data, where the objective function minimizes the total expected propagated delays on selected strings. Recently, Liang et al. (2011) addressed a multi-day ARP by constructing a rotation-tour time-space network with wrap-around maintenance arcs. The authors considered maintenance routing both as a feasibility problem and as an optimization problem, the latter of which incorporated through-values and short connection penalties by adding artificial arcs. Their model is of polynomial space complexity and can be handled by commercial software. In a follow-on paper, Liang and Chaovalitwongse (2011) extended their previous model to consider the weekly aircraft maintenance routing problem by duplicating the daily network multiple times and linking them with proper maintenance arcs. The authors also incorporated fleet assignment decisions within this modeling framework by replicating the same routing network for each type of aircraft, and proposed a variable fixing heuristic to effectively solve the resulting large-scale model.

As alluded above, the ARP is also often formulated as a feasibility problem since the costs associated with aircraft routing are comparatively negligible. Gopalan and Talluri (1998) and Talluri (1998) solved the ARP from a combinatorial perspective. They represented a daily aircraft path by its starting and ending stations in a multi-day maintenance routing problem. The authors developed efficient heuristics to generate qualified routes, and proved that no polynomial-time algorithm exists if the planning horizon extends longer than three days.

2.1.3 Integrated Models Including the ARP

Despite the efforts applied toward solving the ARP as well as other stages of the scheduling process, a sequential optimization process for the overall airline scheduling problem often results in suboptimal solutions, because this methodology overlooks the underlying interrelations and interdependences among consecutive stages. Therefore, it has been of increasing interest to integrate the ARP with other decision-making stages to achieve better solutions. It is worthwhile noting that the size of such integrated models grows rapidly with the number of aircraft types and flight legs, making it difficult to solve even moderately sized problems. Hence, these models are often implemented using sophisticated, specialized algorithms based on column generation and/or Benders decomposition techniques.

Barnhart et al. (1998a) proposed a flight string model (FSM) that considers fleet assignment and aircraft routing simultaneously. The authors defined a string as a sequence of connected flights that begins and ends at maintenance stations, and an augmented string as a string having an additional minimum time for maintenance attached to the end of the last flight. This model attempted to directly assign each aircraft in a fleet to a string in order to minimize the total cost minus through-values. The authors also addressed obtaining a balanced utilization of aircraft within a fleet by introducing subtour elimination constraints in a fleet-specific problem. The model was solved using a branch-and-price algorithm (Barnhart et al., 1998b). Alternatively, Haouari et al. (2009, 2011b), and Mansour et al. (2010) investigated an aircraft fleeting and routing problem for TunisAir, with additional features such as *deadhead flights*, i.e., flights that are scheduled to relocate the aircraft and crews. Their model was simplified by assuming that maintenance is route-independent and that passengers only travel a single leg. The authors formulated a network flow-based model as well as a set partitioning type model, and proposed a heuristic procedure as well as an exact approach to solve the problem to optimality.

From another perspective, since the ARP strongly influences feasibility in the subsequent crew pairing (CP) decisions, there also exists a body of literature on integrating the ARP with the CP. Cohn and Barnhart (2003) proposed an extended crew pairing model that minimizes the total cost for pairings. They delayed the key aircraft decisions regarding *short connections*, i.e., consecutive flights that allow a crew to connect only if the same aircraft serves both the flights, and included them in the crew pairing model. They further argued that only *unique* and *maximal* maintenance-feasible short connections need to be considered, and thus modified their basic integrated model to generate such connections in a preprocessing phase. Moreover, Cordeau et al. (2001), Mercier et al. (2005), Mercier and Soumis (2007), and Mercier (2008) have conducted extensive research on designing different solution approaches. Besides short connections, *restricted connections* (i.e., connections for which a crew pairing has less than the ideal time for changing aircraft) were also introduced so as to enhance the robustness of the integrated model. The authors incorporated column generation within a Benders decomposition scheme, and developed a three-phase solution procedure that progressively imposes integrality restrictions on sets of variables. They further tested the effectiveness of a variety of models and solution methods, and compared these with existing approaches in the literature. The results revealed that using the CP to formulate the master program and the ARP as the subproblem yielded the best overall performance. Most recently, Weide et al. (2010) addressed robustness issues within an integrated aircraft routing and crew scheduling approach. Their aircraft-to-follow-the-crews methodology iteratively solved the crew pairing problem and improves robustness by maximizing the number of restricted connections in the aircraft routing decisions. Their results demonstrated that robustness can be achieved with relatively small cost tradeoffs.

In addition, we note that the stage of schedule design can also be beneficially combined with that of fleet assignment, since the timing of flights and the flight frequencies between pairs of stations bear direct relationships with possible connections and assigned aircraft capacity. Accordingly, Lohatepanont and Barnhart (2004) integrated these two stages by incorporating optional flights within the schedule, with the focus on recapturing spilled demand among the same or close origin-destination pairs. The authors proposed a variable-fixing heuristic that employs row and column generation procedures in order to effectively solve this problem. Furthermore, Sherali et al. (2010) included itinerary-based demands for different fare-classes within their integrated airline schedule design and fleet assignment model, which was further tightened by deriving several classes of valid inequalities for enhancing problem solvability. The authors additionally adopted Benders decomposition for more effectively solving the resulting model. In follow-on research, Sherali et al. (2011) further addressed in their integrated model other important relevant decisions such as retiming flights, balancing schedules, and recapturing demand.

2.1.4 Integrated Models for Fleeting, Routing, and Crew Pairing

Even though numerous papers have addressed enhanced airline fleeting models that accommodate aircraft routing or crew pairing considerations, only two papers deal with models that effectively capture aircraft fleeting, aircraft routing, and crew pairing aspects. Since these two papers are particularly relevant to our work, we describe them in relatively greater detail below.

The first of these is by Sandhu and Klabjan (2007), which presented a model that integrates the fleeting and crew pairing decisions, while representing aircraft maintenance considerations in only an aggregate sense via certain aircraft-count requirements. The authors argued that since the aircraft maintenance routing process is relatively less complicated (and this is true for hub-and-spoke networks), one can neglect detailed routing decisions within the integrated model and simply include some suitable aggregated aircraft-count restrictions. To this end, they designed a novel set of constraints to link the fleet assignment and the crew pairing decision variables that would promote feasible aircraft routings. Yet, as demonstrated by Papadakos (2009), feasible aircraft rotations are not always guaranteed for the resulting solution. The model proposed by Sandhu and Klabjan is based on the traditional time-space network where each node represents an activity or event at a station (arrival or departure of a flight) and each arc represents a flight leg.

In their integrated model, a *forced turn* defines a connection opportunity that is shorter than the minimum sit-time for crews. In such a case, the aircraft must follow the crews and thus force a plane-turn. In addition, an *essential ground arc* refers to a connection arc that is shorter than the minimum sit-time in the *actual-time-space network* (ATN), where each essential arc corresponds to a ground arc in the *ready-time-space network* (RTN), i.e., the ATN in which an additional minimum aircraft turn-time is added to the actual arrival time of each flight. The authors showed that incorporating these essential ground arcs are sufficient to capture all the plane-count requirements. Specifically, for an essential ground arc of a particular aircraft family, the total number of forced turns involved cannot exceed the total sum for this aircraft family of the number of corresponding ground arcs in the RTN, plus the sum of aircraft that take off during the particular related minimum sit-time interval.

To effectively solve the formulated problem, the authors proposed two solution schemes based respectively on using Lagrangian relaxation with column generation and using Benders decomposition. In the former method, they adopted Lagrangian duals for pricing candidate pairings within a constrained shortest path subproblem, where generated pairings were incorporated within the restricted master program, which was then resolved using a subgradient optimization technique, and the process was reiterated until some termination criterion was met. Having thus obtained the fleet assignment decisions, the problem was decomposed into traditional crew pairing problems, one for each fleet type (or aircraft family), which were then solved separately as usual. Alternatively, a Benders decomposition approach (Benders, 1962) was also developed for solving this problem, where the fleet assignment decisions were maintained within the master program, and the crew pairing decisions were incorporated within the subproblem while relaxing integrality restrictions. The proposed methods were tested on four instances having sizes ranging from two fleet families and 205 legs to four fleet families and 942 legs. The authors found that the Lagrangian relaxation approach was more robust in terms of objective value improvements than the Benders decomposition methodology. However, while the computational times for the former method ranged from 15 to 34 CPU hours, the latter required 5 to 19 CPU hours on a cluster of 27 dual 900 MHz Itanium 2 processors running the Red Hat 7.3 operating system.

The second paper that fully integrates the aforementioned three operational stages is by Papadakos (2009), who developed a model that minimizes the total cost of fleet assignment, aircraft maintenance routing, and crew pairing, less the revenues from through-flights. Similar to Sandhu and Klabjan's model, the proposed formulation is also based on a time-space network in which each node represents an activity and each arc represents a connection between consecutive activities. The resulting model adopts a path-based representation that facilitates partitioning the network into separate components for each tail number and crew group. The objective function captures the total cost of routing and pairing, which also implicitly includes the cost for fleet assignment. Since this work is closely related to our problem, we present its mathematical formulation below with some more detailed discussions.

Parameters:

F: set of fleets, indexed by f.

L: set of flight legs, indexed by l.

P^f: set of pairings of fleet type f, indexed by p^f, where p is used as a generic fleet-independent index notation.

R^f: set of routes of fleet type f, indexed by r^f, where r is used as a generic fleet-independent index notation.

S^f: set of leg pairs that can be short-connected for fleet f.

M^f: set of maintenance activity nodes for fleet type f, indexed by m^f, where m is used as a generic fleet-independent index notation.

m^+/m^-: successor/predecessor nodes of m at the same station as for maintenance activity node m.

$\hat{M}^f \subseteq M^f$: set of maintenance activity nodes m of fleet type f, for which the arc (m, m^+) crosses the scheduling horizon.

a_{lp}: binary indicator that equals 1 if flight leg l is in pairing p, and 0 otherwise.

e_{lr}: binary indicator that equals 1 if flight leg l is in route r, and 0 otherwise.

c_{fl}: cost of assigning fleet $f \in F$ to flight leg $l \in L$.

c_p: cost of pairing p.

$c_p^+ = c_p + \sum_{l \in L} c_{fl} a_{lp}$: cost of pairing p plus the cost associated with assigning the appropriate flight legs to pairing p of fleet type f, $\forall p \in P^f, f \in F$.

c_r: cost of route r.

$c_r^+ = c_r + \sum_{l \in L} c_{fl} e_{lr}$: cost of route r plus the cost associated with assigning the appropriate flight legs to route r of fleet type f, $\forall r \in R^f, f \in F$.

e_{mr}^+/e_{mr}^-: binary indicator that equals 1 if maintenance activity m is the initial/terminal maintenance station for route r.

$\hat{e}_r$: integer indicator that equals the number of times that route r crosses the scheduling horizon (excluding terminal maintenance).

n_f: number of available aircraft for fleet type f.

s_f^{ij}: binary indicator that equals 1 if the MR solution allows a short-connection between flight legs i and j for fleet type f, and 0 otherwise.

s_p^{ij}: binary indicator that equals 1 if flight legs i and j are short-connected in pairing p, and 0 otherwise.

s_r^{ij}: binary indicator that equals 1 if flight legs i and j are short-connected in route r, and 0 otherwise.

Decision Variables:

q_m: integer decision variable that counts the number of aircraft on the ground between starting times of m and m^+.

v_r: binary decision variable that equals 1 if route r is adopted, and 0 otherwise.

w_p: binary decision variable that equals 1 if pairing p is adopted, and 0 otherwise.

x_{fl}: binary decision variable that equals 1 if fleet type f is assigned to flight leg l, and 0 otherwise. Note that $x_{fl} = \sum\limits_{r \in R^f} e_{lr} v_r, \forall f \in F, l \in L$.

The model is presented as follows:

$$\text{Minimize} \qquad \sum_{f \in F} \sum_{r \in R^f} c_r^+ v_r + \sum_{f \in F} \sum_{p \in P^f} c_p w_p \tag{2.1}$$

subject to:

$$\sum_{f \in F} \sum_{r \in R^f} e_{lr} v_r = 1, \qquad \forall l \in L, \tag{2.2}$$

$$\sum_{p \in P^f} a_{lp} w_p - \sum_{r \in R^f} e_{lr} v_r = 0, \qquad \forall l \in L, f \in F, \tag{2.3}$$

$$\sum_{p \in P^f} s_p^{ij} w_p - \sum_{r \in R^f} s_r^{ij} v_r \leq 0, \qquad \forall (i,\ j) \in S^f, f \in F, \tag{2.4}$$

$$q_m - q_{m^-} + \sum_{r \in R^f} (e_{mr}^+ - e_{mr}^-) v_r = 0, \qquad \forall m \in M^f, f \in F, \tag{2.5}$$

$$\sum_{m \in \hat{M}^f} q_m + \sum_{r \in R^f} \hat{e}_r v_r \leq n_f, \qquad \forall f \in F, \tag{2.6}$$

$$v_r \in \{0,\ 1\}, \qquad \forall r \in R^f, f \in F, \tag{2.7}$$

$$w_p \in \{0,\ 1\}, \qquad \forall p \in P^f, f \in F, \tag{2.8}$$

$$q_m \geq 0,\ \text{integer}, \qquad \forall m \in M^f, f \in F. \tag{2.9}$$

The objective function (2.1) minimizes the total cost for aircraft routing, crew pairing, as well as, implicitly through the defined cost coefficients, for fleet assignment. Constraint (2.2) requires that each flight leg is assigned to exactly one route. Constraint (2.3) enforces that each flight leg is assigned to exactly one crew pairing for a particular fleet if and only if the flight leg is assigned to that fleet. Constraint (2.4) ensures that a short-connection for a pairing of fleet type f is possible only if the connection is assigned to the same tail number for that fleet. Constraint (2.5) imposes flow conservation for each fleet type at every associated maintenance station. Constraint (2.6) ensures that for each fleet type, the total number of aircraft in use does not exceed its fleet size. Constraints (2.7)-(2.9) represent logical restrictions on the decision variables.

For solving the model effectively, the author focused on implementing Benders decomposition while treating the crew pairing problem within the Benders subproblem, and while formulating the Benders master program to incorporate the fleet assignment and maintenance routing decisions, where both these decomposed problems were solved using a branch-and-price strategy. In addition, *Pareto-optimal* or nondominated Benders cuts were generated using the technique of Magnanti and Wong (1981) in order to accelerate the solution process and to address degeneracy in the Benders subproblem. An alternative approach in which the roles of crew pairing and aircraft routing are reversed within the Benders decomposition framework was also explored, but was found to be not as effective as the former strategy.

Since the formulated model contains a large number of constraints as well as variables, the author proposed a Benders decomposition algorithm that accommodates aircraft maintenance routing within the master program and crew scheduling within the subproblem. The crew network in the subproblem is separable according to each fleet type, and the crew pairing for each individual sub-network was solved using column generation, where the subproblem involved finding a constrained shortest path in the crew-connection network. Additionally, in order to speed up the crew pairing subproblem solution, the labels for the total flying time and the flying time during the past 24 hours were not considered so as to reduce the number of labels. Furthermore, a deepest-cut pricing rule was adopted to determine the nonbasic variable to be inserted into the basis during the column generation process, which was demonstrated to be three times faster on average than the traditional Dantzig's rule. The subproblem solution was used to generate Benders feasibility and optimality cuts, which were iteratively incorporated within the master program until a specified termination criterion was met. In this process, due to the fact that a set partitioning structure is highly degenerate, the author discriminated among alternative optimal dual solutions as per the technique proposed in Magnanti and Wong (1981) in order to generate Pareto-optimal (nondominated) cuts by solving the subproblem using a *core point* (a point located in the relative interior of the convex hull of the underlying defining discrete set). Note that since the core point is only approximated in the implementation, the generated cuts were not strictly Pareto-optimal but were yet found to be sufficiently strong to accelerate the convergence process. Likewise, the Benders master program that accommodates maintenance routing in addition to fleet assignment decisions was also solved using a column generation approach. The column generation subproblem in this context served to find a constrained shortest path in the aircraft-connection network, which was mostly the same as the crew-connection network except that it had additional maintenance arcs. Since the entire problem is a mixed-integer program (MIP), the proposed Benders decomposition approach was embedded within a branch-and-bound algorithm in which the fleet assignment was determined first and then branching was performed with respect to the resultant flight legs in the maintenance routing subproblem for each fleet type. The proposed search process adopted a depth-first strategy and terminated once a first integral solution was detected.

Papadakos tested his model on seven instances. In all cases, he considered six aircraft fleets and a number of flight legs ranging from 214 to 705 (the number of aircraft was not indicated). He reported that the CPU times ranged from 0.35 to 27.8 hours on a single computer having a 2.4 GHz single-core AMD Athlon processor with 2 GB RAM, and running the 64-bit Linux kernel version 2.6.11.

2.2. Description of Some Selected Relevant Works

In this section, we first present a comprehensive review of the airline scheduling process as described by Klabjan (2005). Next, we discuss in detail some principal models related to the aircraft routing problem, which is the focus of our work in Chapter 3. Finally, we describe in detail integrated models that include the aircraft routing problem as one of its components, similar to our contribution in Chapter 4.

2.2.1 Airline Scheduling Problem

In the book *Column Generation* compiled by GERAD, Klabjan (2005) presented a comprehensive review on large-scale models adopted for the airline scheduling problem. Three common solution techniques: branch-and-price, Lagrangian relaxation, and Benders decomposition were described in detail. Furthermore, the author discussed different stages of the airline planning problem with basic models, references, and recent advances in these areas.

Following the main topic of the book, the solution approaches that are often adopted in airline scheduling problems include a *delayed column generation* procedure, where attractive columns are generated within the context of a branch-and-price algorithm. During the column generation process, a restricted master problem is solved first, and then the subproblem is evaluated to identify a column that has a negative reduced cost. If none exists, the solution is declared to be optimal to the continuous relaxation; otherwise, the newly generated column is appended to the master program and the procedure is reiterated. It is worth noting that solving the subproblem is usually expensive; comparatively, obtaining a *constrained shortest path* using reduced costs is computationally efficient in practice (Desrosiers et al. (1995) and Desaulniers et al. (1997)).

Branch-and-price, an offshoot of the branch-and-bound algorithm, combines the branching strategy and the delayed column generation method at each node. To maintain a more balanced tree, it is proposed to branch on whether or not a node s immediately follows another node r, i.e., either only the arc (r, s) or all arcs from r except (r, s) are removed from the network. Also, due to the intensive effort required for solving the LP relaxation, it is proposed to adopt this approach only for the first several nodes in a depth-first search until a feasible integer solution is obtained. A detailed discussion of the branch-and-price concept is given in Barnhart et al. (1998b).

Second, Lagrangian relaxation is also a commonly adopted technique where so-called complicating constraints are dualized, i.e., accommodated within the objective function. The remaining specially structured constraints simplify the resulting problem, and in practice, a subgradient-based algorithm often serves as an effective solver for the underlying Lagrangian dual problem.

Third, Benders decomposition (Benders, 1962) can be applied to mixed-integer programs that have a dual angular structure, which yields block-diagonal constraints after fixing certain integer variables. An optimal dual solution to the resulting LP, when it exists, yields a Benders cut, also referred to as an optimality cut, which is then appended to the relaxed master program for the next iteration. Likewise, whenever the resulting LP is infeasible, a feasibility cut is generated based on an unbounded (extreme) direction for the dual problem.

From the viewpoint of the airline decision-making timeline, the service plan development, which usually starts 12 months before operations, is the first phase on which all other subsequent decisions are based. From nine months prior to the day of operations to a few weeks prior, the flight scheduling, fleet assignment, aircraft routing (also called maintenance routing), and crew scheduling operational decisions are made, usually sequentially, all of which comprise the core for airline scheduling. Additionally, during the last few weeks, some minor changes are made based on, for example, crew bidline and plane swapping operations. Furthermore, the actual scheduling that takes place on any particular day enacts minor adjustments and disruption management. The latter step includes three aspects: aircraft recovery, crew recovery, and passenger reaccommodation. We will mainly focus on the aforementioned passenger-side planning problems.

The fleet assignment problem (Abara, 1989; Hane et al., 1995) is based on a flight time-space network in which a node (u, i) represents an activity of leg i at station u. The activities i are arranged in increasing order of their times t_i. Flight arcs $((u, i), (v, j))$, ground arcs $((u, i), (u, j))$, and wrap-around arcs are all directed arcs that appropriately link the different activities.

Inputs:

- A list of flights given by origin/destination pairs and departure/arrival times.
- A set of aircraft fleets, each with its fleet size and the specified seating capacity.

Output:

- Assignment of fleets to flights that yields an optimal estimated revenue.

Parameters:

A: set of all flight arcs.

V: set of nodes.

K: set of fleet types.

M: set of flights in the air at a certain time et.

W: set of ground arcs at a certain time et.

$I(v)/O(v)$: set of flight arcs to/from node v.

$i(v)/o(v)$: ground arc to/from node v.

b_k: number of aircraft in fleet k.

c_{ik}: cost of assigning fleet k to flight leg i.

Decision Variables:

x_{ik}: binary variable that equals 1 if flight leg i is assigned to fleet k, and 0 otherwise.

y_{gk}: nonnegative integer variable that represents the flow on ground arc g for fleet type k.

The fleet assignment model (FAM) is then given as follows:

$$\textbf{FAM:} \quad \text{Minimize} \quad \sum_{i\in A}\sum_{k\in K} c_{ik}x_{ik} \tag{2.10}$$

subject to:

$$\sum_{k\in K} x_{ik} = 1, \qquad \forall i \in A, \tag{2.11}$$

$$\sum_{i\in O(v)} x_{ik} - \sum_{i\in I(v)} x_{ik} + y_{o(v)k} - y_{i(v)k} = 0, \qquad \forall v \in V, k \in K, \tag{2.12}$$

$$\sum_{g\in W} y_{gk} + \sum_{i\in M} x_{ik} \leq b_k, \qquad \forall k \in K, \tag{2.13}$$

$$x \text{ binary}, \tag{2.14}$$

$$y \geq 0, \ \text{integer}. \tag{2.15}$$

The objective function (2.10) is to minimize the total assignment costs. Constraint (2.11) ensures that each leg is assigned to exactly one fleet. Constraint (2.12) maintains the flow conservation of each type of aircraft. Constraint (2.13) assures that the number of aircraft in use for each type does not exceed the corresponding fleet size. Constraints (2.14) and 2.15 represent logical restrictions on the decision variables.

The fleet assignment model often serves as a foundational basis for other expanded integrated airline models. Clarke et al. (1996), and Rushmeier and Kontogiorgis (1997) incorporated constraints of aircraft maintenance and crew requirements within the FAM, and Barnhart et al. (1998a) introduced explicit aircraft routes within the basic model. Also, the model can be extended to accommodate flight departure time decisions, as discussed by Rexing et al. (2000), Desaulniers et al. (1997), and Bélanger et al. (2006).

The FAM model needs an augmentation to capture the revenues from multi-leg itineraries since the decision of a fleet assignment to a flight leg can potentially impact the revenues on the remaining flight legs in an itinerary whenever there are spilled passengers. Therefore, an alternative *passenger-mix model* (Kniker, 1998) is constructed under the assumptions of a single fare for each itinerary and no recaptured passengers. This model, abbreviated PMIX, optimizes the number of booked passengers for each itinerary.

Additional Parameters:

A: set of flight legs.

P: set of itineraries.

f_p: fare for itinerary $p \in P$.

$\tilde{C}_k$: seat capacity of fleet type k.

C_i: available seat inventory of leg i.

D_p: unconstrained demand for itinerary $p \in P$.

Additional Decision Variables:

w_p: nonnegative integer variable that counts the number of booked passengers for itinerary $p \in P$.

PMIX: Maximize
$$\sum_{p \in P} f_p w_p \tag{2.16}$$

subject to:

$$\sum_{p:\ i \in p} w_p \leq C_i, \qquad \forall i \in A, \tag{2.17}$$

$$w_p \leq D_p, \qquad \forall p \in P, \tag{2.18}$$

$$w \geq 0,\ \text{integer}. \tag{2.19}$$

The objective function (2.16) maximizes the total revenue. Constraint (2.17) requires that the total number of passengers booked on any flight leg should not exceed the seating capacity of the assigned fleet type. Constraint (2.18) imposes the specified limits on the number of booked passengers, and Constraint (2.19) requires the w-variable to be integer-valued.

Furthermore, the *origin-destination fleet assignment model* (OD-FAM) combines the FAM and the PMIX together, where the only change is to replace Constraint (2.17) by

$$\sum_{p:\ i \in p} w_p \leq \sum_{k \in K} \tilde{C}_k x_{ik}, \qquad \forall i \in A. \tag{2.20}$$

The OD-FAM is not as easy to solve as the FAM. The branch-and-price algorithm, proposed by Barnhart et al. (2002) and Kniker (1998) required several CPU hours to find the first integer solution, not to mention to solve the problem entirely. However, the model became more tractable after several enhancements were applied. The FAM can also incorporate schedule design decisions, or aircraft routes and schedule design together.

Aircraft routing is the next stage that follows fleet assignment. The solution to this problem assigns a sequence of flights to each individual aircraft (*tail number*), while accommodating maintenance checks (principally, the ones called type A and type B checks as discussed previously). Also, airlines attempt to achieve a balanced utilization rate of aircraft, which is sometimes incorporated within the model.

Aircraft routing can be further partitioned into two stages. Several weeks prior, generic aircraft routes that satisfy only the A-checks are generated in the *aircraft rotation problem*. Furthermore, the actual tail numbers are assigned to each flight by solving the *aircraft assignment problem*, in which the B-checks are accommodated.

Most of the ARP models are formulated using the presented network-flow framework; however, it is worth noting that Gopalan and Talluri (1998), and Talluri (1998) have proposed a combinatorial framework in which the aircraft rotation problem is modeled as finding an Eulerian tour. Furthermore, we note that the aircraft routing problem that needs to be solved on the same day of operations has also been studied by several researchers. Usually, the planned schedule needs real-time adjustments due to unexpected events. Short aircraft recoveries can be modeled as a minimum-cost network optimization model using a multicommodity network-flow framework, while constraints for maintenance regulations must be added in long disruption scenarios. In both cases it is very difficult to solve the problem to optimality due to the complexity of the underlying models and the urgency of time. Therefore, heuristics are often employed to determine quick but suboptimal solutions. Detailed descriptions of ARP models will be presented in Section 2.2.2.

The crew scheduling problem involves a pair of problems: the *crew pairing problem* (CP) and the *crew rostering problem*. In this context, a *duty* is a sequence of flights for a crew on a particular day, and a *duty pairing* (or simply a *pairing*) is a sequence of daily duties that originates from and terminates at the same base during a given time period (usually a month). The pairing process requires finding a minimum-cost crew assignment that satisfies a variety of regulations, such as the minimum sit-time, the maximum elapsed time, the maximum flying time on duty, the 8-in-24 rule, the maximum time away from base, and other more complex duty time regulations. Also note that the cost function for a pairing, which is nonlinear, depends on the maximum of the following three elements: the elapsed time for the pairing, the total cost of duties within the pairing, and the guaranteed pay per duty multiplied by the number of duties. The complexity of regulations and the nonlinearity of the cost function makes the problem extremely difficult to solve. Alternatively, if the crew is offered a fixed salary by the airline company, then the objective simply becomes to minimize the total number of crew members.

Additional Parameters:

P: set of all pairings.

S_{cb}: set of all pairings that start and end at crew base cb.

l_{cb}/u_{cb}: the lower/upper bound on the number of crews at cb.

c_p: cost of pairing $p \in P$.

Additional Decision Variables:

x_p: binary decision variable that equals 1 if pairing p is selected, and 0 otherwise.

$$\textbf{CP:} \quad \text{Minimize} \quad \sum_{p \in P} c_p x_p \tag{2.21}$$

subject to:

$$\sum_{p:\ i \in p} x_p = 1, \qquad \forall i, \tag{2.22}$$

$$l_{cb} \leq \sum_{p \in S_{cb}} x_p \leq u_{cb}, \qquad \forall cb, \tag{2.23}$$

$$x_p \in \{0, 1\}, \qquad \forall p \in P. \tag{2.24}$$

The objective function (2.21) minimizes the total cost of pairings. Constraint (2.22) ensures that each flight leg is assigned to exactly one pairing. Constraint (2.23) is optional, and restricts the number of crews utilized at each crew base, and Constraint (2.24) represents logical binary restrictions on the decision variables.

Due to the intractability of the above model, the crew pairing optimization is usually divided into three steps: the daily problem, the weekly exceptions problem (or the weekly problem), and the *dated* problem where a daily schedule is assumed to be recurrent everyday. Similar to the above mentioned models, the problem also employs the flight network structure; however, not every path is feasible since it must satisfy the mandated regulatory requirements.

In the branch-and-price solution technique employed to solve this problem, branching with respect to *follow-on* flights is the most commonly adopted strategy. Another branching rule is called *time-line branching* (Klabjan et al., 2002), which branches on the flights whose connection times are shorter or greater than a given time period. After the crew pairing problem is solved, the specific identified duties are assigned to each individual crew member by solving the crew rostering problem, as described below. However, it is worth noting that in some airline companies, a bidline process or a preferential bidding is carried out instead, which allows crew members to bid for their favorite duties.

Additional Parameters:

K: set of all crew members.

S: set of rosters.

n_i: number of crew members that are required by flight i.

c_s^k: cost of assigning crew member k to roster s.

Additional Decision Variables:

x_s^k: binary variable that equals 1 if roster s is selected for crew member k, and 0 otherwise.

Rostering: Minimize

$$\sum_{k \in K} \sum_{s \in S} c_s^k x_s^k \tag{2.25}$$

subject to:

$$\sum_{k \in K} \sum_{s \in S:\ i \in s} c_s^k x_s^k \geq n_i, \qquad \forall i, \tag{2.26}$$

$$\sum_{s \in S} x_s^k = 1, \qquad \forall k, \tag{2.27}$$

$$x_s^k \in \{0, 1\}, \qquad \forall k, s. \tag{2.28}$$

The objective function (2.25) minimizes the total cost for assigning crew members to rosters. Constraint (2.26) ensures that each flight has at least the required number of crew members. Constraint (2.27) assures that each crew member is only assigned to exactly one roster. Finally, Constraint (2.28) imposes binary restrictions on the decision variables.

2.2.2 Models for Aircraft Routing Decisions

The *aircraft routing problem* (ARP) has been extensively studied both as an individual problem and as a component that is integrated with other decision stages. In this subsection, we focus on some principal models that solely address aircraft routing decisions. In the next subsection, we shall discuss integrated models that include the ARP.

Feo and Bard (1989)

Feo and Bard (1989) proposed a model that combines flight scheduling and maintenance station location. The authors formulated a minimum-cost, multicommodity network flow model, and solved it using a two-phase heuristic. In this paper, the objective is to establish a maintenance base planning framework in which the fixed plus variable costs for performing type A checks are minimized for given schedule maintenance requirements.

In the network construction, each node represents a city in a day of the horizon. Also, a sequence of flight legs serviced by a single aircraft in a day is identified by its origin and destination, i.e., an OD pair, which is represented by an edge. Note that the interim stops are unimportant. Such a graph, which contains a total of $n_d \cdot n_c$ nodes and $n_d \cdot n_p$ edges, is large but sparse, where n_d, n_c, and n_p are defined below.

The network can be of infinite or finite time horizon. It can be argued that this graph is Eulerian and therefore removing any cycle does not affect the Eulerian property. Each aircraft (*tail number*) represents a separate commodity and the flights are provided as input. Also, the model assumes that the flow is cyclic and repeats itself every n_d days. Furthermore, the authors split each city-day node into two nodes connected by two arcs, one of which is a maintenance arc having a capacity of p_j aircraft and the other is a non-maintenance arc having infinite capacity. The model also assumes that the maximal interval between two sequential maintenances is four days.

The notation for this proposed model is described as follows.

Network Parameters:

n_d: length of the planning time horizon.

n_c: number of cities in the OD schedule.

n_p: number of aircraft in the fleet.

E: set of edges in the network defined by the OD schedule ,where $|E| = n_d \cdot n_p$.

i: index for aircraft, where $i = 1, \ldots, n_p$.

j, k: indices for cities, where $j, k = 1, \ldots, n_c$.

d: index for days, where $d = 1, \ldots, n_d$.

$j(d)$: city j at the end of day d; node designation.

p_j: capacity of maintenance base in city j.

f_j: fixed cost of setting up a maintenance base in city j.

g_j: unit cost of performing maintenance in city j.

Decision Variables:

$x_{ij(d)k(d+1)}$: binary decision variable that equals 1 if aircraft i is in city j at the end of day d and in city k at the end of day $d+1$, and 0 otherwise.

$w_{ij(d)}$: binary decision variable that equals 1 if aircraft i gets maintenance in city j at the end of day d, and 0 otherwise.

δ_j: binary decision variable that equals 1 if city j is designated as a maintenance base, and 0 otherwise.

y_{id}: integer variable indicating the number of days remaining before aircraft i gets maintenance, where this number is recorded at the end of day d prior to maintenance.

This problem is formulated as follows:

$$\text{Minimize} \quad \sum_{j=1}^{n_c} \big(f_j \delta_j + \sum_{i=1}^{n_p} \sum_{d=1}^{n_d} g_{j(d)} w_{ij(d)} \big) \tag{2.29}$$

subject to:

$$\sum_{j(d-1)} x_{ij(d-1)k(d)} = \sum_{j(d+1)} x_{ik(d)j(d+1)}, \qquad \forall i, k\left(d\right), \tag{2.30}$$

$$\sum_{i=1}^{n_p} x_{ij(d)k(d+1)} = 1, \qquad \forall d, \left(j\left(d\right), k\left(d+1\right)\right) \in E, \tag{2.31}$$

$$\sum_{i=1}^{n_p} w_{ij(d)} \leq p_j \delta_j, \qquad \forall d, j\left(d\right), \tag{2.32}$$

$$w_{ij(d)} \leq \sum_{k(d-1)} x_{ik(d-1)j(d)}, \qquad \forall i, j\left(d\right), \tag{2.33}$$

$$y_{i,d+1} - y_{id} \leq 4\big(\sum_{j(d)} w_{ij(d)}\big) - 1, \qquad \forall i, d, \tag{2.34}$$

$$x_{ij(d)k(d+1)} \in \{0,1\}, \qquad \forall i, j, k, d, \tag{2.35}$$

$$w_{ij(d)} \in \{0,1\}, \qquad \forall i, j, d, \tag{2.36}$$

$$\delta_j \in \{0,1\}, \qquad \forall j, \tag{2.37}$$

$$y_{id} \in \{0,1,2,3\}, \qquad \forall i, d. \tag{2.38}$$

The first term in the objective function (2.29) represents the fixed cost of operating the facilities and the second term sums up the variable costs for unit maintenances. Constraint (2.30) ensures the conservation of flow for each aircraft. Constraint (2.31) requires that each OD pair is covered by exactly one aircraft in a day. Constraint (2.32) restricts the number of aircraft that can receive maintenance at a city in a day. Constraint (2.33) is another set of conservation inequalities, which asserts that only aircraft arriving at a city in a day can possibly get maintenance there. Constraint (2.34) guarantees that all the aircraft get proper maintenance within four consecutive days. Constraints (2.35)-(2.38) record logical restrictions, where y_{id} is restricted to take on nonnegative integral values no greater than three because this pertains to the evening before the maintenance.

The above formulation is a large-scale mixed-integer program that is difficult to solve directly. Instead, the authors proposed to decompose the model by aircraft type. Although this significantly reduces the complexity of the approach, the model still remains intractable at that time.

A more realistic approach adopted by the planning group at *American Airlines* is presented below. They did not use the model developed above but start by generating feasible tail number assignments (routes). As inputs, their model requires initial location and maintenance conditions as well as a set of feasible routes. It is virtually impossible to enumerate the complete route set; rather, AA randomly generates several thousand possible routes *a priori*.

Additional Parameters:

n: number of paths generated.

m: number of OD legs to be covered where $m = n_p \cdot n_c \cdot n_d$.

c_j: cost associated with path j.

a_{ij}: binary indicator that equals 1 if OD leg i is covered by path j, and 0 otherwise.

Decision Variables:

x_j: binary variable that equals 1 if path j is chosen; 0 otherwise.

The currently used model is formulated as the following set partitioning problem:

$$\text{Minimize} \qquad \sum_{j=1}^{n} c_j x_j \tag{2.39}$$

subject to:

$$\sum_{j=1}^{n} a_{ij} x_j = 1, \qquad \forall i = 1, \ldots, m, \tag{2.40}$$

$$x_j \in \{0, 1\}, \qquad \forall j = 1, \ldots, n. \tag{2.41}$$

Here, each path corresponds to a particular aircraft. If a generated path is valid, it is ascribed a fixed cost $\bar{c}$; else, if invalid, it is given a high cost $\bar{C}$, where $\bar{C} \gg \bar{c}$. Then, with all valid paths, the optimal solution value is necessarily $n_p \bar{c}$. However, deadheads may be required if the optimal solution value is greater. Usually, such set partitioning problems are solved using cutting planes with implicit enumeration; however, the authors suggested a heuristic instead. We refer the reader to their paper for details. Modern-day branch-and-cut software (such as CPLEX) might be able to better handle such model formulations.

Daskin and Panayotopoulos (1989)

In the paper by Daskin and Panayotopoulos (1989), the authors proposed an assignment-based formulation for assigning aircraft to routes in a hub-and-spoke network. The routes defined in this problem were simplified to an out-and-back structure, i.e., "each route originates at a single hub, visits a number of other cities (usually only one) and returns to the hub". The timetable of flights was treated as being exogenous and defined by the departure/arrival time and ground service time. However, it is worth mentioning that, in their model, the time periods were generated with each departure that follows an arrival. Also, the expected profit for a route served by an aircraft was specified. Note that not all the routes were necessarily covered in the solution since deleting some may increase the profit (see the example provided in their paper).

Parameters:

I: set of routes, indexed by $i \in I$.

J: set of aircraft, indexed by $j \in J$.

K: set of time periods, indexed by $k \in K$.

D_i: departure time of route i from the hub.

$A_i{}'$: effective arrival time of route i at the hub.

P_{ij}: profit due to assigning route i to aircraft j.

N_{jk}: set of routes that could possibly utilize aircraft j during the period k.

Decision Variables:

X_{ij}: binary variable that takes the value 1 if route i is assigned to aircraft j, and 0 otherwise.

This aircraft routing problem (ARP) is formulated as follows:

$$\textbf{ARP:} \quad \text{Maximize} \quad \sum_{i \in I} \sum_{j \in J} P_{ij} X_{ij} \tag{2.42}$$

subject to:

$$\sum_{j \in J} X_{ij} \leq 1, \qquad \forall i, \tag{2.43}$$

$$\sum_{i \in N_{jk}} X_{ij} \leq 1, \qquad \forall j, k, \tag{2.44}$$

$$X_{ij} \in \{0, 1\}, \qquad \forall i, j. \tag{2.45}$$

Specifically, the objective 2.42 is to maximize the total revenue by assigning aircraft to routes according to the published timetable. Constraint (2.43) requires that each route is assigned to at most one aircraft. Constraint (2.44) ensures that during each time period k any aircraft is assigned to at most one route. Constraint (2.45) imposes logical restrictions on the decision variables. To ensure the smallest number of time period constraints, the paper suggested to initiate a new time period only when there is a departure after an arrival. The maximum number of time periods generated in this fashion is $|I|$, while the total number of Constraints (2.43) and (2.44) is $|I|\,(|J| + 1)$.

This formulation has a special set packing structure that can be readily transformed into a network structure by using variables x_{ijk}. The authors proposed a Lagrangian-relaxation based heuristic for the problem, but with today's technology and by exploiting the network structure, problems of the scale considered can be solved easily.

Clarke, Johnson, Nemhauser, and Zhu (1997)

Clarke et al. (1997) discussed the aircraft routing problem (ARP) that determines the routes of each aircraft in a given fleet. Given the fleet assignment decision, the authors formulated the problem for each fleet type as an *acyclic traveling salesman problem* (ATSP) with the objective to maximize through-values less operational costs. Certain maintenance constraints were also included.

A *broken rotation* refers to the situation where aircraft in the same fleet cover separate subsets of its assigned flights. Such a rotation is said to have a *continuity break* if the continuous cycle of all the flight legs assigned to a particular fleet is broken. As per airline rules, these breaks are forbidden in the model, which leads to the acyclic traveling salesman problem. Moreover, there may also exist some *locked rotations* where separate rotations for a single fleet do not have any common station. The authors assumed that such cases do not arise since otherwise the fleet would be reassigned to avoid the lock. Furthermore, the *service index* is defined as the longest break between two consecutive services. A rotation is called feasible only if it is free of continuity breaks, and the service index lies below the specified service period.

The model due to Clarke et al. (1997) is based on a time-space network, where each arc connection is associated with a revenue. Usually, a *through-value* is assigned to those arcs whose connection time is no longer than 90 minutes. Furthermore, after node aggregation, the network flow might admit some formerly impossible connections where a later arrival connects to an earlier departure at a station; however, such a connection is prohibited by assigning it a negative infinity revenue.

In the directed graph $N(V, A)$, an Euler tour visits all the arcs exactly once. Moreover, the Euler tour corresponds to a Hamiltonian cycle in the line digraph of N, denoted by D, whose nodes are the arcs in N and where arc (i, j) is constructed if i and j are respectively the inbound and outbound flights for the same station in N. Also, through-values are assigned to the arcs in D.

A *service violation* path contains segments whose service index exceeds the given threshold. The set of all minimum service violation paths are denoted by P^k for maintenance type $k \in K$. In the model, there are two types of maintenance checks that are considered: *avionic* check and *routine* check; but in most of the related literature, we only see one type called the A-check.

Network Parameters:

A: set of directed arcs, indexed by $i, j \in A$.

$h(i)/t(i)$: head/tail node of arc $i \in A$.

$S \subseteq A$: subset of A.

v_{ij}: through-value of flight j following flight i; it equals 0 if i or j is a ground arc.

$C = \{i, j \in A \text{ s.t. } h(i) = t(j), i \neq j\}$.

$P' \subseteq P^k$: subset of arcs in P excluding the last arc.

$f(i)$: the follower of $i \in P'$.

Decision Variables:

x_{ij}: binary variable that equals 1 if there is a connection from i to j, and 0 otherwise.

Problem ARP is formulated as follows:

$$\textbf{ARP:} \quad \text{Maximize} \quad \sum_{(i,j)\in C} v_{ij} x_{ij} \tag{2.46}$$

subject to:

$$\sum_{\substack{j:\ t(j)=h(i),\\ j\neq i}} x_{ij} = 1, \qquad \forall i \in A, \tag{2.47}$$

$$\sum_{\substack{i:\ h(i)=t(j),\\ i\neq j}} x_{ij} = 1, \qquad \forall j \in A, \tag{2.48}$$

$$\sum_{\substack{i\in S,\ j\in A\setminus S:\\ h(i)=t(j)}} x_{ij} \geq 1, \qquad \forall S \subseteq A \text{ with } 2 \leq |S| \leq |A| - 2, \tag{2.49}$$

$$\sum_{\substack{i \in P',\ j \in A \setminus f(i): \\ h(i)=t(j)}} x_{ij} \geq 1, \qquad \forall P \in P^k, k \in K, \tag{2.50}$$

$$x_{ij} = \{0, 1\}, \qquad \forall i, j \in A. \tag{2.51}$$

The objective function (2.46) is to maximize the total revenue of all connections. Constraints (2.47) and (2.48) ensure that every arc in N is traversed exactly once. Constraint (2.49) is the continuity break constraint, which is also known as the subtour elimination constraint. Constraint (2.50) eliminates service violation paths and ensures that proposed paths are maintenance-feasible, and Constraint (2.51) enforces binary logical restrictions.

During a preprocessing step, *arc aggregation* and *node aggregation* were performed to simplify the model. Specifically, whenever a node has an in-degree of one, its in-arc and out-arc are aggregated into a super arc, and the intermediate node is deleted. The through-value on such a super arc is defined by its last inbound and first outbound flights. This procedure ensures that every remaining node has a degree of two or more. Moreover, since the through-value is based on flight connections, the ground arcs at a station can be eliminated, along with corresponding departure/arrival nodes consolidated. This procedure generates a super-node for each station. As stated above, infeasible connections at a super-node are prohibited by assigning through-values of negative infinity.

The authors proposed a Lagrangian relaxation based heuristic to solve the developed model. They also adopted conventional subgradient optimization to solve the Lagrangian dual formulation. Due to the exponential number of constraints, only the identified violated constraints were dualized.

Desaulniers, Desrosiers, Dumas, Solomon, and Soumis (1997)

The daily aircraft routing and scheduling problem (DARSP) addressed one of the most important decisions for airline companies since it involved a high proportion of airline operational costs. Given a set of aircraft fleets, a one-day flight schedule with departure time-windows and durations, and fleet- and flight-dependent costs, the objective was to maximize profits while satisfying coverage, flow balance, and maintenance constraints. Note that, although the flight departure time has a narrow window, it was assumed that departures are time-independent. Desaulniers et al. (1997) proposed a set partitioning type formulation and a multicommodity network flow formulation for DARSP and developed sophisticated mathematical programming algorithms to determine optimal solutions.

Parameters:

N: set of operational flight legs.

K: set of aircraft types.

n^k: number of available aircraft of type $k \in K$.

Ω^k: set of feasible schedules for aircraft of type $k \in K$, indexed by p. Let $p = 0$ denoted the empty schedule for an aircraft.

c_p^k: anticipated profit if schedule $p \in \Omega^k$ is assigned to an aircraft of type $k \in K$.

a_{ip}^k: binary constant that equals 1 if the schedule covers flight leg $i \in N$, 0 otherwise.

S: set of stations.

$S^k \subseteq S$: subset of stations having facilities to maintain aircraft of type $k \in K$.

o_{sp}^k/d_{sp}^k: binary constants that equal 1 if schedule $p \in \Omega^k$ starts/ends at station $s \in S^k$, and 0 otherwise.

Decision variables:

θ_p^k: binary decision variable that equals 1 if schedule p is assigned to an aircraft of type k, where $p \in \Omega^k \setminus \{0\}, k \in K$, and 0 otherwise.

θ_0^k: nonnegative integer decision variable that equals the number of unused aircraft of type $k \in K$.

Set Partitioning Model:

$$\text{Maximize} \qquad \sum_{k \in K} \sum_{p \in \Omega^k} c_p^k \theta_p^k \tag{2.52}$$

subject to:

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip}^k \theta_p^k = 1, \qquad \forall i \in N, \tag{2.53}$$

$$\sum_{p\in\Omega^k} \left(d_{sp}^k - o_{sp}^k\right)\theta_p^k = 0, \qquad \forall k \in K, s \in S^k, \tag{2.54}$$

$$\sum_{p\in\Omega^k} \theta_p^k = n^k, \qquad \forall k \in K, \tag{2.55}$$

$$\theta_p^k \geq 0, \text{ integer}, \qquad \forall k \in K, p \in \Omega^k. \tag{2.56}$$

The objective function (2.52) maximizes the total anticipated profit by summing the profits pertaining to the selected routes. Constraint (2.53) requires that each flight leg is covered exactly once. Constraint (2.54) represents the flow conservation of each fleet at each station. Constraint (2.55) puts a cap on the number of useable aircraft of each type. Finally, Constraint (2.56) requires the variables to be nonnegative integers.

A column generation technique was used to solve this formulation, which decomposes the problem into a restricted master problem and a subproblem. Here, the master program considers a relatively small subset of feasible aircraft routes that satisfy Constraints (2.53)-(2.56) in the LP relaxation sense, and generates corresponding dual solutions for solving the subproblem. The subproblem is a longest path problem with time-windows, and its network structure is given by $G^k = (V^k, A^k)$, where V^k is the set of nodes that is comprised of five types: *source* $o(k)$; *sink* $d(k)$; *initial station* S_1^k; *final station* S_2^k; and *flight* $N^k \subseteq N$; and where the arc set A_k includes seven types: *empty* denoted by $OD^k = (o(k), d(k))$; *source* denoted by $OS_1^k = (o(k), s), s \in S_1^k$; *sink* denoted by $S_2D^k = (s, d(k)), s \in S_2^k$; *schedule start* denoted by $S_1N^k = (s, j), s \in S_1^k, j \in N^k$; *schedule end* denoted by $NS_2^k = (i, s), i \in N^k, s \in S_2^k$; *turn* denoted by $NN^k = (i, j), i, j \in N^k$; and *short turn* denoted by NQN^k. Let a_i^k/b_j^k be the earliest/latest time at which leg i/j can begin, l_i^k the duration of leg i, t_{ij}^k the normal connection time between leg i and j, and s_{ij}^k shorter connection time with $s_{ij}^k < t_{ij}^k$. Hence, for a normal turn arc, $a_i^k + l_i^k + t_{ij}^k \leq b_j^k$ holds; while for a short turn arc, $a_i^k + l_i^k + s_{ij}^k \leq b_j^k$ holds and a penalty cost is incurred.

The schedule generated by the subproblem must be feasible and have a positive marginal revenue. Define the revenue on arc $(o(k), d(k))$ as $-e_o^k, k \in K$; the profit of flight leg j on every pertaining leaving arc as $r_i^k - e_i^k$, where r_i^k and e_i^k respectively denote the expected revenue and cost; the short turn penalty cost as q_{ij}^k; and a large fixed operational cost on every source arc as M. Then the marginal (reduced) profit on a schedule $p \in \Omega^k$ can be represented as

$$\bar{c}_p^k = c_p^k - \sum_{i \in N^k} \alpha_i a_{ip}^k - \sum_{s \in S^k} \sigma_s^k \left(d_{sp}^k - o_{sp}^k \right) - \beta^k,$$

where α_i, σ_s^k, and $\beta_k, \forall i \in N, k \in K, s \in S^k$, are dual variables that correspond to Constraints (2.53)-(2.55).

Note that the definition of a flight leg can be extended to include a through-flight, which represents a series of consecutive legs that are assigned to the same aircraft. In addition, the authors also presented a time-constrained multicommodity network flow formulation, as introduced below.

Additional parameters:

d_{ij}^k: the duration of activities on arc $(i, j) \in A^k$, depending on the aircraft type $k \in K$.

Decision Variables:

X_{ij}^k: integer flow variable indicating the number of type k aircraft using arc (i, j), where $k \in K, (i, j) \in A^k$.

T_i^k: a time variable for aircraft type $k \in K$ at node i such that, if $i \in N^k$, then this denotes the departure time of leg i within the time interval $[a_i^k, b_i^k]$; otherwise, it is fixed at 0.

Multicommodity Network Flow Model:

$$\text{Maximize} \quad \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij}^k X_{ij}^k \tag{2.57}$$

subject to:

$$\sum_{k \in K} \sum_{j:(i,j) \in A^k} X_{ij}^k = 1, \qquad \forall i \in N, \tag{2.58}$$

$$\sum_{i:(i,s) \in NS_2^k} X_{is}^k - \sum_{j:(s,j) \in S_1 N^k} X_{sj}^k = 0, \qquad \forall k \in K, s \in S^k, \tag{2.59}$$

$$\sum_{s \in S_1^k} X_{o(k),s}^k + X_{O(k),d(k)}^k = n^k, \qquad \forall k \in K, \tag{2.60}$$

$$\sum_{i:(i,j) \in A^k} X_{ij}^k - \sum_{j:(j,i) \in A^k} X_{ji}^k = 0, \qquad \forall k \in K, j \in V^k \setminus \{o(k), d(k)\}, \tag{2.61}$$

$$\sum_{s \in S_2^k} X_{s,d(k)}^k + X_{o(k),d(k)}^k = n^k, \qquad \forall k \in K, \tag{2.62}$$

$$a_i^k \leq T_i^k \leq b_i^k, \qquad \forall k \in K, i \in V^k, \tag{2.63}$$

$$X_{ij}^k(T_i^k + d_{ij}^k - T_j^k) \leq 0, \qquad \forall k \in K, (i,j) \in A^k, \tag{2.64}$$

$$X_{ij}^k \geq 0, \text{ integer}, \qquad \forall k \in K, (i,j) \in A^k. \tag{2.65}$$

The objective function (2.57) maximizes the total profit derived from the network. Constraint (2.58) requires that each flight leg is covered exactly once. Constraint (2.59) ensures the flow conservation for each fleet at each station. Constraints (2.60) and (2.62) assure that the number of scheduled aircraft of each fleet type do not exceed the maximum number available. Constraint (2.61) represents the node conservation for the network. Note that Constraints (2.60) and (2.61) together make Constraint (2.62) redundant. Constraints (2.63) and (2.64) define the time-window requirement and establish the relationship between the flight assignment and its departure time. The last constraint (2.65) requires the flow variables to be nonnegative integers.

The integer node-arc and node-path formulations presented are equivalent, and the node-path formulation can be derived from the node-arc formulation by performing Dantzig-Wolfe decomposition. This permits the development of branching strategies that are compatible with the column generation technique.

Sriram and Haghani (2003)

Sriram and Haghani (2003) described a model for the aircraft maintenance routing problem, in which they included aircraft re-assignments when possible. Due to the complexity of their model, the authors proposed a random search-based approach and presented benchmark results for their algorithm using several test scenarios.

In this paper, a weekly domestic cyclic schedule served by various types of fleet was studied, in which both type A checks and type B checks were taken into account. Since all the maintenance checks are done during night times, a daily flight plan, or a trip for each aircraft was thus represented by its origin and destination stations (OD pair) respectively at the beginning and end of a day. Given the fleet assignment, the model determines the flight and maintenance plans for each individual aircraft under mandatory maintenance requirements, with an objective function that minimizes the maintenance costs plus the aircraft re-assignment penalties. The formulation takes the form of a minimum-cost multi-commodity network flow problem on a directed Eulerian graph, which is implied from the fact that the in-degree equals the out-degree for every node.

Parameters:

r: subscript that identifies different trips that share the same OD pair, i.e., trips that have the common origin and destination of a day but stop at different intermediate stations.

n_p: number of aircraft in the fleet, indexed by $i = 1, \ldots, n_p$.

n_c: number of stations in the network, indexed by $j, k = 1, \ldots, n_c$.

n_d: number of days in the planning horizon, indexed by $d = 1, \ldots, n_d$, where $n_d + 1 \equiv 1, n_d + 2 \equiv 2$, and $n_d + 3 \equiv 3$.

j_d: station j on day d in the network.

$j_{d-1}k_dr$: arc that connects station j on day $d-1$ to station k on day d through route r.

$G(k_d)$: set of stations (on different routes r) that are connected to station k_d.

$F(k_d)$: set of stations (on different routes r) that station k_d is connected to.

L: set of all arcs in the graph of the OD schedule.

N: set of all stations in the graph of the OD schedule.

g_{ij}: cost of type A maintenance for aircraft i at station j.

h_{ij}: proportional cost (see explanations below) of type B checks for aircraft i at station j.

p_j: number of aircraft that can be simultaneously maintained at station j.

$c_{ij_{d-1}k_dr}$: cost of assigning aircraft i to the arc $j_{d-1}k_dr$. This can be interpreted as a re-assignment penalty depending on a previous assignment being altered.

Decision Variables:

w_{ij_d}: binary variable that equals 1 if aircraft i receives type A check at station j on day d, and 0 otherwise.

z_{ij_d}: binary variable that equals 1 if aircraft i receives type B check at station j on day d, and 0 otherwise.

$x_{ij_{d-1}k_dr}$: binary variable that equals 1 if aircraft i is assigned to arc $j_{d-1}k_dr$, and 0 otherwise.

This model is presented as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n_p}\sum_{j=1}^{n_c}\sum_{d=1}^{n_d}(g_{ij}w_{ij_d} + h_{ij}z_{ij_d}) + \sum_{i=1}^{n_p}\sum_{ij_{d-1}k_dr\in L} c_{ij_{d-1}k_dr}x_{ij_{d-1}k_dr} \tag{2.66}$$

subject to:

$$\sum_{j_{d-1}r\in G(k_d)} x_{ij_{d-1}k_dr} - \sum_{j_{d+1}r\in F(k_d)} x_{ik_dj_{d+1}r} = 0, \qquad \forall i, k_d,\ d = 1,\ldots,n_d, \tag{2.67}$$

$$\sum_{i=1}^{n_p} x_{ik_dj_{d+1}r} = 1, \qquad \forall k_dj_{d+1}r \in L,\ d = 1,\ldots,n_d, \tag{2.68}$$

$$\sum_{i=1}^{n_p} w_{ij_d} \le p_j, \qquad \forall j_d,\ d = 1,\ldots,n_d, \tag{2.69}$$

$$w_{ij_d} - \sum_{k_{d-1}r\in G(j_d)} x_{ik_{d-1}j_dr} \le 0, \qquad \forall i, j_d,\ d = 1,\ldots,n_d, \tag{2.70}$$

$$\sum_{j=1}^{n_c}\sum_{d=m}^{m+3} w_{ij_d} \ge 2, \qquad \forall i,\ m = 1,\ldots,n_d, \tag{2.71}$$

$$\sum_{j_d\in N} z_{ij_d} = 1, \qquad \forall i, \tag{2.72}$$

$$z_{ij_d} - \sum_{k_{d-1}r\in G(j_d)} x_{ik_{d-1}j_dr} \le 0, \qquad \forall i, j_d,\ d = 1,\ldots,n_d, \tag{2.73}$$

$$\sum_{k_2r\in G(j_3)} x_{ik_2j_3r} = 1, \qquad \forall i, j_3, \tag{2.74}$$

$$x_{ij_dk_{d+1}r},\ w_{ij_d},\ z_{ij_d} \in \{0,1\}. \tag{2.75}$$

The idea behind this model formulation is to essentially represent a node-arc network flow that (i) maintains flow balance at each node; (ii) has a capacity for each flight and maintenance arc; (iii) imposes maintenance feasibility at each station, and (iv) keeps track of the number of days between two successive maintenance operations. In addition, the subscript

r denotes different routes that share the same OD pair. In Constraint (2.67), the summation terms represent flows on all the arcs to and from a node, respectively. Moreover, the problem requires that each and every aircraft is on service everyday, which therefore makes Constraint (2.74) an equality.

The objective function (2.66) minimizes the total cost of type A checks, type B checks, plus the penalties for assigning inappropriate aircraft to OD trips. Constraint (2.67) ensures the flow balance at each node of the network. Constraint (2.68) enforces that each OD trip is served by exactly one aircraft. Constraint (2.69) restricts the number of aircraft that can be serviced simultaneously at a maintenance station. Constraint (2.70) requires that an aircraft undergoes a type A check at a station only if it overnights there. Constraint (2.71) forces that each aircraft must undergo type A checks every four days. Constraint (2.72) assures that each aircraft passes a type B check once every study cycle; however, as asserted by the authors, this "does not necessarily mean that the aircraft has to undergo a type B maintenance check every seven days", i.e., only an opportunity to do so is provided. Also, due to the same reason, only a proportion of the cost of type B checks (based on an assumed expected ratio of actual to potential maintenance services) is included in the objective function. Constraint (2.73), similar to Constraint (2.70), guarantees that an aircraft cannot possibly undergo the type B check at a particular station unless the day's OD pair ends there. Constraint (2.74) states that, at any given timeline (e.g., Day-3 in above formulation), any aircraft can only be assigned to exactly one OD pair. Constraint (2.75) imposes logical restrictions on the decision variables.

The above formulation can be simplified by separating it by fleet type, and by deleting the rarely binding Constraint (2.69). However, the remaining problem is still too large to handle within a reasonable computational time. Therefore, the authors proposed a solution approach that performs random search and depth-first search together. The approach assigns a random starting node to an aircraft, generates a cycle greedily for the aircraft from that node, eliminates all the served OD pairs if the cycle is valid, and restarts the process for the next aircraft from another randomly chosen node. This step was iterated for a number of times, and the minimum-cost assignment is recorded along the way. Benchmark results revealed a gap of at most 5% between the heuristic and optimal solutions. Rather than keeping track of the days, the authors also formulated the problem based on the real flight hours of an aircraft, which is similar to the above-presented formulation.

Sarac, Batta, and Rump (2006)

Sarac et al. (2006) addressed a short-term aircraft maintenance routing problem for daily operational decision-making. A tailored branch-and-price algorithm was designed to efficiently solve the problem, as supported by computational results.

Instead of taking several days as the study time horizon, the authors focused on operational maintenance routing decisions within a day in order to better respond to dynamic environmental effects such as inclement weather, unexpected equipment failures, and emergency maintenance requests. The model took into account different types of maintenances, i.e., maintenances for resident and non-resident components, which require distinct maintenance frequencies and man-hours. Maintenance requests for a single aircraft were batched together, where the legal remaining flying time was computed as the minimum of that of its components. The model only incorporated the *high-time* aircraft that have accumulated flying hours up to a predefined threshold. These aircraft need to be routed to maintenance stations for receiving necessary checks.

The objective of the daily aircraft maintenance routing problem is to minimize the total daily maintenance costs of a fleet while satisfying mandatory FAA requirements on regular checks and other operational constraints. Specifically, this model minimized the total *cushion time*, defined as the unutilized but legal remaining flying hours of an aircraft before it is maintained, and therefore, maximized the utilization of the total legal flying time (called *green time*) of a fleet of high-time aircraft. Note that non-high-time aircraft can be potentially incorporated within the model, but servicing these aircraft earlier than necessary is not considered worth the time and cost.

A connection network was adopted as the underlying structure of the model, in which a node represents a flight leg, and an arc represents a possible connection at a station, while accommodating a sufficient turn-time. Dummy source and sink nodes were also connected to appropriate starting and ending flight legs, with respective flight durations of zero and that of the inbound flight. With this directed graph, a set partitioning formulation was proposed as described below.

Parameters:

K: set of aircraft, indexed by $k \in K$.

o/t: dummy source/sink node.

A: set of connection arcs.

N: set of flight legs, indexed by $i \in N$.

R_k: set of feasible routes, indexed by $j \in R_k$, that are generated for aircraft k.

M: set of maintenance types, indexed by $m \in M$.

S_m: set of overnighting stations, indexed by $s \in S_m$, where maintenance m can be performed.

t_i: duration of flight i.

c_j^k: cost of selecting route j for aircraft k.

τ_k: legal remaining flying hours of aircraft k.

a_m^k: maintenance man-hours needed to perform maintenance m for aircraft k.

b_m^k: binary indicator that equals 1 if aircraft k needs maintenance type m, and 0 otherwise.

d_{js}^k: binary indicator that equals 1 if route j served by aircraft k ends at overnight station s, and 0 otherwise.

λ_{js}: binary indicator that equals 1 if flight leg i arrives at overnight station m, and 0 otherwise.

γ_{ji}^k: binary indicator that equals 1 if route j served by aircraft k contains flight leg i, and 0 otherwise.

L_{ms}: available man-hours for maintenance type m at overnight station s.

Z_{ms}: number of available opportunities for maintenance m at overnight station s.

Decision Variables:

y_j^k: binary variable that equals 1 if route j of aircraft k is selected, and 0 otherwise.

This model is formulated as follows:

$$\text{Minimize} \qquad \sum_{k \in K} \sum_{j \in R_k} c_j^k y_j^k \tag{2.76}$$

subject to:

$$\sum_{j\in R_k} y_j^k = 1, \qquad \forall k \in K, \tag{2.77}$$

$$\sum_{k\in K}\sum_{j\in R_k} \gamma_{ji}^k y_j^k = 1, \qquad \forall i \in N, \tag{2.78}$$

$$\sum_{k\in K}\sum_{j\in R_k} a_m^k d_{js}^k y_j^k \leq L_{ms}, \qquad \forall s \in S_m,\ m \in M, \tag{2.79}$$

$$\sum_{k\in K}\sum_{j\in R_k} b_m^k d_{js}^k y_j^k \leq Z_{ms}, \qquad \forall s \in S_m,\ m \in M, \tag{2.80}$$

$$y_j^k \in \{0,1\}, \qquad \forall j \in R_k,\ k \in K. \tag{2.81}$$

The objective function (2.76) minimizes the total cost or the unutilized legal flying hours for a fleet. Constraint (2.77) ensures that each aircraft is assigned to exactly one route, while Constraint (2.78) enforces that each flight leg is covered by exactly one aircraft. Constraint (2.79) requires that the total man-hours needed for a maintenance type at a particular station should not exceed the available number of man-hours. Constraint (2.80) imposes an upper bound on the maintenance opportunities for each aircraft type at each station. Constraint (2.81) represents logical restrictions on the decision variables.

The set partitioning formulation was solved by a branch-and-price procedure. During the preprocessing, primary attention was focused on the nodes having zero or one in- and out-degrees. Nodes with zero in-degree must be starting flights, and a zero out-degree implies an ending flight at an overnighting station. Moreover, a node with an in-degree of one can be combined with its preceding node since the single connection arc must be covered by the same aircraft. Similarly, a node with an out-degree of one can be aggregated with its succeeding node.

An initial feasible solution was derived by a heuristic that uses the first-in-first-out (FIFO) strategy to generate connections for each aircraft. Once a valid route was obtained for an aircraft, all of its flight legs were eliminated from the graph. However, this scheme does not always guarantee a feasible solution. Another approach was proposed where each flight coverage constraint was relaxed and built into the objective function with a penalty. The associated artificial variables were eliminated later along the process of solving the restricted master program (RMP). Interested readers are referred to Savelsbergh and Sol (1998) for details of this approach.

Problem RMP was solved using a column generation approach that prices the reduced costs for the $|K|$ aircraft individually. If none of the prices is negative, then optimality is achieved. The structure of the pricing problem is that of a constrained shortest path (CSP) problem with a restriction on the legal remaining flying hours. Problem CSP was solved by a generalized permanent labeling algorithm as proposed by Desrochers and Soumis (1989), where this problem is known to be NP-hard (Handler and Zang, 1980).

Branching on *follow-ons* can possibly fail in this problem; therefore, the authors proposed two other branching strategies. The first one, called aircraft-specific follow-on, ordered the aircraft according to the remaining legal flying hours, and calculated the value of $\sum_{j\in R_k^{(n,f)}} y_j^k$ for a pair of flight legs (n, f). New nodes were generated for forcing each aircraft to cover the connection, or none of the aircraft covers the connection, or the two flights are flown nonconsecutively. The second strategy, namely, the aircraft-specific ending flight leg method, focused on which aircraft can fly the ending flight. The ending flight that is shared by most aircraft was selected to branch on, and new nodes were generated for each scenario where the particular flight is covered by a possible aircraft.

Alternative strategies for route generation and selection were also discussed. Essentially, routes can be generated independently for each aircraft, or dependently. The former discipline motivates the generation of many promising columns, but there may exist identical routes for different aircraft, which results in low efficiency. The latter discipline requires that disjoint routes are sequentially generated for a pre-sorted list of aircraft. Although this strategy avoids redundant routes, it needs several runs to cover all the flight legs and to provide sufficient good-quality candidates.

Finally, numerical studies on a series of test scenarios were provided, which revealed that the dependent route generation strategy with a premature termination of the pricing subproblems performs most effectively for all problem sizes, independent of the branching strategy implemented.

Lan, Clarke, and Barnhart (2006)

Lan et al. (2006) built robustness into their models that minimize passenger disruptions by optimally routing aircraft and intelligently selecting departure times. Specifically, their work addressed a robust aircraft maintenance routing problem and a flight schedule retiming model. We will focus on the former, which is a deterministic mixed-integer program with the objective of minimizing expected total propagated delays as explained below.

Delays and cancellations of flights disrupt planned aircraft routing and crew scheduling, and also cause customer inconveniences, all of which lead to significant losses for both the airline company and passengers. In reality, however, most conventional aircraft routing models only focus on minimizing the *planned* cost, but do not account for the *realized* cost, which additionally includes the costs of disruptions. Moreover, the conventional methodology dealing with schedule disruptions is to re-optimize the schedule in real-time whenever necessary. However, it is suggested that cost reduction can be more effective if possible delays and cancellations are considered in the planning stage by using historical data.

As pointed out by Klabjan et al. (2002) and Cordeau et al. (2001), through-revenues are difficult to evaluate and are financially less important; therefore, the maintenance routing problem can be reduced to a feasibility problem that requires type A checks only. Moreover, rather than maximizing the total revenue of routing, it is natural for the robust aircraft maintenance routing model (RAMR) to concentrate on minimizing the total expected delays under restrictions of maintenance routing.

Flight delays are classified as two types: *propagated delays* due to prior flights of the assigned aircraft and *nonpropagated delays* caused by other reasons, the latter of which are therefore independent of routing decisions. Let TAD_i denote the total arrival delay for flight i, which is given by the sum of propagated delays and delays caused by other independent reasons. Let $slack_{ij}$ be the difference between the planned turn-time and the minimum turn-time between flights i and j, then the propagated delay when both flights are served by the same aircraft is defined as

$$PD_{ij} = \max\{TAD_i - slack_{ij},\ 0\}.$$

Since the independent delay can hardly be controlled, the model focused on absorbing the propagated delay by the slack time to the extent possible.

The historical data for delays and cancellations were extracted from the *Airline Service Quality Performance* (ASQP) database. The authors found that a log-normal distribution best fits the data; therefore, once the parameters are determined, the expected delay can be calculated offline.

Since the fleet assignment is determined *a priori*, the problem can be solved independently for each fleet type. As presented below, the proposed formulation is based on the string model of aircraft routing developed by Barnhart et al. (1998a).

Parameters:

S: set of feasible strings.

F: set of daily flight legs.

F^+/F^-: set of flight legs that originate/terminate at a maintenance station.

G: set of ground arcs and overnight arcs.

S_i^+/S_i^-: set of strings beginning/ending with flight leg i.

pd_{ij}^s: delay propagated from flight leg i to j, given that flight legs i and j are in string s.

a_{is}: binary indicator that equals 1 if flight leg i is in string s, and 0 otherwise.

$y_{i,d}^+/y_{i,d}^-$: integer indicator that equals the number of aircraft on the ground after/before flight leg i departs.

$y_{i,a}^+/y_{i,a}^-$: integer indicator that equals the number of aircraft on the ground after/before flight leg i arrives.

r_s: number of times that string s crosses the count-time.

p_g: number of times that ground arc g crosses the count-time.

N: number of aircraft available for the fleet type.

Decision Variables:

x_s: binary decision variable that equals 1 if string s is selected, and 0 otherwise.

y_g: integer decision variable that counts the number of aircraft on the ground at maintenance station g.

$$\textbf{RAMR:} \quad \text{Minimize} \qquad E\Big(\sum_{s\in S}\big(\sum_{(i,j)\in s} pd_{ij}^s\big)x_s\Big) \tag{2.82}$$

subject to:

$$\sum_{s \in S} a_{is} x_s = 1, \qquad \forall i \in F, \tag{2.83}$$

$$\sum_{s \in S_i^+} x_s - y_{i,d}^- + y_{i,d}^+ = 0, \qquad \forall i \in F^+, \tag{2.84}$$

$$\sum_{s \in S_i^-} x_s + y_{i,a}^- - y_{i,a}^+ = 0, \qquad \forall i \in F^-, \tag{2.85}$$

$$\sum_{s \in S} r_s x_s + \sum_{g \in G} p_g y_g \leq N, \tag{2.86}$$

$$y_g \geq 0, \text{ integer}, \qquad \forall g \in G, \tag{2.87}$$

$$x_s \in \{0, 1\}, \qquad \forall s \in S. \tag{2.88}$$

The objective function (2.82) minimizes the total expected propagated delays of the selected strings. The stochasticity of the objective is eliminated by noting that

$$\begin{aligned} E\Big[\sum_{s \in S} \big(\sum_{(i,j) \in s} pd_{ij}^s\big) x_s\Big] &= \sum_{s \in S} \Big(x_s \cdot E\big[\sum_{(i,j) \in s} pd_{ij}^s\big]\Big) \\ &= \sum_{s \in S} \Big(x_s \cdot \sum_{(i,j) \in s} E\big[pd_{ij}^s\big]\Big) \end{aligned}$$

Constraint (2.83) requires that each flight leg is covered by exactly one string. Constraints (2.84) and (2.85) ensure the flow balance at each node by equating the numbers of aircraft departing from and arriving at a station. Constraint (2.86) counts the number of aircraft in the system at a given count-time and requires this not to exceed the specified fleet size. Constraints (2.87) and (2.88) represent logical restrictions on the decision variables.

The formulated model was solved using a branch-and-price approach, i.e., a branch-and-bound process with column generation being used to solve the LP relaxation at each node. It is worth mentioning that the pricing subproblem cannot be modeled as a constrained shortest-path problem in this case because the delay of a connection depends on the string that it belongs to, and hence a reduced cost cannot be assigned to a connection arc. Therefore, a heuristic was employed in which a candidate string was generated before its reduced cost was calculated. Moreover, the branching strategy used emphasized *follow-on* flights. Specifically, branching was performed based on whether or not a flight leg follows the current one in a string, whence the problem can be resolved within the framework of the connection digraph with corresponding arcs eliminated.

Liang, Chaovalitwongse, Huang, and Johnson (2011)

Liang et al. (2011) presented an interesting new model for aircraft maintenance routing (AMR) using the time-space network, which they demonstrated offers a more effective formulation than the flight string model (Barnhart et al., 1998a).

AMR, which is the same as Problem ARP referred to above, is a strategic-level decision-making problem that determines the flight assignment for each aircraft while satisfying maintenance restrictions set by FAA and the airline. The model incorporates a daily flight schedule with maintenance checks performed at least every $|D|$ days at some designated stations by constructing a time-space network. The time-space network is defined by a timeline on one axis and station locations on the other axis. Hence, each station has an associated timeline. On this domain, the nodes represent departure or arrival events. The arcs are partitioned into ground arcs, flight arcs, and overnight arcs to respectively represent aircraft staying at one station, flights between stations, and overnight halts or maintenance checks. Node aggregation and island isolation were applied to simplify the graph as in Hane et al. (1995).

Network Parameters:

$D = \{1, \ldots, |D|\}$: set of possible days between two consecutive maintenance operations, indexed by $d \in D$ that represents the day in the rotation-tour network.

N: set of nodes (events) in the network, indexed by $n \in N$.

F: set of daily flights, indexed by $f \in F$.

M: set of maintenance stations, indexed by $m \in M$.

K: size of the scheduled fleet.

Q_m: maximum number of maintenances allowed per day at station m, where $m \in M$.

$\alpha^+_{fdn}/\alpha^-_{fdn}$: binary indicator that equals 1 if flight f on day d starts/ends at node n, and 0 otherwise.

$\beta^+_{mdn}/\beta^-_{mdn}$: binary indicator that equals 1 if maintenance arc at station m starts/ends on day d, and 0 otherwise.

l^+_n/l^-_n: ground arc before/after node n.

C^f: set of profitable connections.

C^s: set of short connections.

F^a/F^d: set of arrival/departure flights.

H: set of additional arcs (including penalty arcs, touching arcs, and connection arcs) within a day, indexed by $h \in H$.

c_h: cost for arc h; specifically, $c_h \geq 0$ for penalty arcs, and $c_h = 0$ for touching and through-value arcs (due to incomplete cost/profit information).

c_{md}: maintenance cost at station $m \in M$ on day $d \in D$.

$\gamma^+_{hdn}/\gamma^-_{hdn}$: binary indicator that equals 1 if arc h at day d starts/ends at node n, and 0 otherwise.

Decision Variables:

x_{fd}: binary decision variable that equals 1 if flight f is flown on day d in the rotation-tour network, and 0 otherwise.

y_{hd}: binary decision variable that equals 1 if arc h at day d is included in the maintenance solution, and 0 otherwise.

z_{md}: integer decision variable that equals the number of aircraft in maintenance at station m at the end of day d.

w_l: integer variable that denotes the number of aircraft on the ground arc l.

The *rotation-tour network model* (RTNM) proposed by Liang et al. is based on a $|D|$-day time-space network $G(N, E)$, where $|D|$ is the maximum number of days allowed between the two consecutive maintenances. Ground arcs and flight arcs are the same as in the traditional time-space network, while the maintenance arcs start at the end of each day at a station and end at the beginning of the same station's timeline (*time-reversed arcs*). Unlike the traditional one-day time-space network, the RTNM has a $|D|$-day duration with time-reversed arcs existing only at maintenance stations. Note that all the maintenance arcs end at the beginning of the maintenance station's timeline, which guarantees at most a $|D|$-day long flight sequence after a maintenance service.

While RTNM seeks to find a feasible rotation tour for AMR, the rotation-tour network optimization model (RTNOM) proposed by the authors is an extended formulation that incorporates the profit/cost of flight connections. Through-value arcs are introduced in this model to account for the profit (negative cost) for every profitable connection, while *touching arcs* are zero-cost arcs that include other non-profitable but feasible transitions. This model can be further extended to handle multiple profitable connections. In this fashion, $|C^f|$ additional through-value arcs as well as $|F^a|+|F^d|$ touching arcs (in and out) are constructed.

Also, for modeling short connections, *penalty arcs* were created with suitable penalty costs. In particular, for every arrival flight at a station, a set of departure flights that form short connections was identified and were linked with corresponding penalty arcs. Note that the penalty also depends on the short time of connections. Formally, $|C^s|$ additional penalty arcs for short connections and $|F^a| + |F^d|$ additional zero-cost arcs were incorporated.This yields the following model:

RTNOM:

$$\text{Minimize} \quad \sum_{h\in H}\sum_{d\in D} c_h y_{hd} + \sum_{m\in M}\sum_{d\in D} c_{md} z_{md} \tag{2.89}$$

subject to:

$$\sum_{d\in D} x_{fd} = 1, \qquad \forall f \in F, \tag{2.90}$$

$$\begin{aligned}\sum_{f\in F}\sum_{d\in D}\alpha^+_{fdn}x_{fd} + \sum_{m\in M}\sum_{d\in D}\beta^+_{mdn}z_{md} + \sum_{h\in H}\sum_{d\in D}\gamma^+_{hdn}y_{hd} + w_{l^+_n} = \\ \sum_{f\in F}\sum_{d\in D}\alpha^-_{fdn}x_{fd} + \sum_{m\in M}\sum_{d\in D}\beta^-_{mdn}z_{md} + \sum_{h\in H}\sum_{d\in D}\gamma^-_{hdn}y_{hd} + w_{l^-_n}, \qquad \forall n \in N,\end{aligned} \tag{2.91}$$

$$\sum_{d\in D} z_{md} \leq Q_m, \qquad \forall m \in M, \tag{2.92}$$

$$\sum_{m\in M}\sum_{d\in D} d \cdot z_{md} \leq K, \tag{2.93}$$

$$x_{fd} \in \{0,1\}, \qquad \forall i \in F, d \in D, \tag{2.94}$$

$$y_{hd} \in \{0,1\}, \qquad \forall h \in H, d \in D, \tag{2.95}$$

$$z_{md} \in \{0,1,\ldots,Q_m\}, \qquad \forall m \in M, d \in D, \tag{2.96}$$

$$w_n^+, w_n^- \geq 0, \qquad \forall n \in N. \tag{2.97}$$

Constraint (2.90) requires that each flight is covered once in the rotation-tour network solution. Constraint (2.91) assures that at each node, the number of inbound aircraft equals the number of outbound ones. Constraint (2.92) ensures that the number of aircraft maintained at each station does not exceed its capacity. Constraint (2.93) restricts the total number of aircraft in service to be no more than its fleet size. Finally, Constraints (2.94)-(2.97) represent logical restrictions on the decision variables.

The total number of variables in RTNOM is $|D|(|F|+|M|+|H|)+|N|$, the total number of constraints is $|F|+|N|+|M|+1$, and the total number of nonzero entries in the problem matrix is $|D|(3|F|+2|H|+4|M|)+2|N|$. Because $O(|N|) = |F|$ and $O(|H|) = |F|^2$, the space complexity of RTNOM is $O(|D||F|^2)$, which is a great saving compared with the space complexity of $O(2^{|F|})$ in FSM. However, the number of constraints in RTNOM is larger than that in FSM, but this can be reduced drastically by preprocessing. In fact, FSM can be viewed as a Dantzig-Wolfe decomposition of RTNOM.

Gopalan and Talluri (1998)

Gopalan and Talluri (1998) addressed the aircraft maintenance routing (MR) problem using a different perspective from other researchers. Since the optimal solution derived for the previous decision-making stage of fleet assignment does not guarantee feasible routes for each tail number, the authors focused on generating valid maintenance routings for each single aircraft. Therefore, instead of formulating a traditional mixed-integer program, they solved a three-day maintenance routing problem within the framework of graph theory. However, since the relevant costs are not associated with the arcs, the resulting feasible solutions are not necessarily optimal.

In this paper by Gopalan and Talluri (1998), the authors considered type A checks that consume three to ten hours and take place every few days, and type C checks that are performed over a longer time period. They proposed a polynomial-time algorithm for solving the three-day routing problem within an infinite-horizon and a finite-horizon model. However, as pointed out in the follow-on research by Talluri (1998), the four-day routing problem was proven to be NP-hard.

The authors also assumed that the same flight schedule is repeated daily. To represent the movement of each tail number, the concept of a *line of flying* (LOF) was introduced to demonstrate the start and end station of a particular aircraft in a day. However, LOFs were treated as invariant along the time axis only in the infinite-horizon case; in the finite-horizon case where type C checks are involved, LOFs pertaining to each tail number can be different from one another.

Additionally, it was assumed that only some of the stations (where this set is denoted by M) can perform type A checks with no capacity limits per night. Therefore, as required by FAA regulations, every tail number must visit a maintenance station at most every k days, which is referred to as a k-day maintenance routing. The scenario with $k = 3$ was discussed in this paper and a polynomial-time algorithm was proposed. Moreover, it was assumed that only one of the maintenance stations is equipped to perform type C checks, with a capacity limit of one aircraft per night. This assumption is reasonable since the type C check requires more rigorous standards than the type A check. The enforced constraint required that the entire fleet visits the particular maintenance station every n days, where n is the size of the fleet. However, there exist situations in which some tail number can never reach the type C check station. This is called a *locked rotation*, which can be possibly unlocked by swapping aircraft at some common station during a given day.

A directed graph $G = (V, E)$ was constructed, where V represents the overnighting stations and E represents the LOFs for aircraft, with $n \equiv |E|$. The set of nodes V were further partitioned into two subsets: the stations M that are equipped with type A check facilities, and the stations $N \equiv V \backslash M$ that are not. Then, as can be proved, the three-day maintenance routing sought an Euler tour in G with at most two nodes from N appearing in succession. Such a tour is named a *three-day maintenance Euler tour (3-MET)*. Note that an Euler tour guarantees visiting each node once every n days, and hence it automatically satisfies the requirement of type C checks.

Define m^o_{jM} as the number of arcs going out of node j to maintenance stations, and let m^i_{jN} be the number of arcs coming into node j from non-maintenance stations. A necessary condition for a 3-MET states that

$$m^o_{jM} \geq m^i_{jN}, \qquad \forall j \in N.$$

To find a valid 3-MET in G, a graph G' was constructed by splitting each node $j \in N$ and its appended arcs into two nodes j' and j'', where j'' only has inflows from maintenance stations and outflows to non-maintenance stations, and node j' has the opposite types of arc flows. In addition, $m^o_{jM} - m^i_{jN}$ artificial arcs were added from node j'' to node j' in G'. Then G' has the property that any Euler tour visits no more than two nodes of N in succession, and it can be proved that there exists a 3-MET in G if and only if an Euler tour exists in G'.

The algorithm used to find an Euler tour in a graph is a classical polynomial-time routine as given by Fleury (1883), and the transformation from G to G' can also be achieved in polynomial time; therefore, a polynomial-time algorithm for 3-MET results. Note, however, that the k-day MET existence problem is NP-complete.

In the finite-horizon model, the routes for each aircraft can vary from day to day; however, the n aircraft must visit the type C check maintenance station overnight once within the n-day study period. Note that the LOFs for each day were given as input. A layered directed graph $G = (V, E)$ was adopted that has n arcs between every pair of successive days in the n-day horizon. The objective is to decompose G into n edge-disjoint paths such that these paths represent valid maintenance routes for each tail number.

The proposed algorithm based on the split graph G' operated over two phases. In Phase I, a unit inflow was appended to each of the type C check stations on day k, where the latter set of nodes is denoted by b_k. Also, a *super-terminus* node T was added to G' with n inflow arcs to T from terminating LOFs on the last day. Assigning a unit capacity to each arc, a network flow problem was solved to determine the flows from all the nodes in b_k to T. After removing all the arcs with integral flows, the procedure moved to the next phase.

Similar to the method adopted in Phase I, a *super-source* node S was added to the remaining graph with n outflow arcs to the first-day nodes. Also, each node in b_k was ascribed a unit demand. A network flow problem was then solved again, which yielded a feasible path from S to each node of b_k. By combining the solutions from Phase I and Phase II at the maintenance stations, routes were formed for each individual aircraft. Since the two phases were performed on the split graph G', the 3-MET was automatically enforced. It is worth mentioning that the existence of an edge decomposition is equivalent to the existence of a Phase I solution.

In conclusion, the entire maintenance routing solver involved three procedures: the *unlocker*, the *M-N improver*, and the *router*. The first two can be viewed as preprocessors, and the last one has been presented above. We will therefore briefly present the preprocessing procedures next.

The unlocker attempted to untie each LOF subtour by swapping aircraft at possible connections during a day in order to create new LOFs. If this method failed, the fleet types for some flight legs, or even for overnighting aircraft, must be changed in order to unlock the LOF-graph. Note that determining the existence of an unlocker solution is actually NP-complete, and the proposed methods are only heuristic that might overlook inherent opportunities.

Following this, the M-N improver swapped pairs of edges to ensure that every node $j \in N$ satisfies the condition that the number of outflow arcs to nodes M is greater than or equal to the number of inflow arcs from nodes N. Particularly, for a node j with $m^o_{jM} < m^i_{jN}$, the procedure selected an inward edge e from another node j_1 along with another swap candidate edge e' that originates from a node of M. A swap was performed if e' terminates at a node of M, or at a node of N that satisfies $m^o_{kM} - m^i_{kN} \geq 1$. It is important to note that such swaps are not permitted unless they do not create locked rotations. Also, note that such swaps will not hurt through-flights because through-flights are combined together and are represented by a single flight arc.

2.2.3 Integrated Models with Aircraft Routing

Apart from the aforementioned models that treat aircraft routing as an individual problem, there also exist a number of aircraft routing models that respectively involve the upstream decision stage of fleet assignment and the downstream decision stage of crew pairing, both of which are described in the sequel. Moreover, we also present sophisticated solution schemes accompanying these large-scale models in this subsection.

Barnhart, Boland, Clarke, Johnson, Nemhauser, and Shenoi (1998a)

The aircraft routing decision stage has obvious interrelationships with fleet assignment; therefore, Barnhart et al. (1998a) introduced the *flight string model* (FSM) in order to stress the synergy of integrating these two stages. In this model, a *string* is defined as a sequence of connected flights that begins and ends at maintenance stations, and that is flow-balanced and maintenance-feasible, i.e., maintenance requirements satisfiable. Furthermore, they defined an *augmented string* as a string that additionally has the minimum time necessary to perform maintenance attached to the end of the last flight in the string. Because such a model contains millions of strings for a moderate-size flight schedule, the authors proposed a branch-and-price approach to solve it.

The input for the fleet assignment model is comprised of a schedule of flight legs, a set of aircraft (fleets) and the associated fleet-specific operational cost, as well as maintenance requirements and the minimum turn-time for different fleets. The output is the assignment of available aircraft to the different flight legs. Opportunity costs due to over booking are also sometimes included in the objective function; moreover, a negative cost of through-flights (Clarke et al., 1997) is additionally incorporated as passengers are willing to pay extra for continuing with the same aircraft at the station for connection. On the other hand, given a flight schedule for a single fleet, the aircraft routing model seeks the minimum-cost set of aircraft routing plans under the constraints of flight coverage, fleet count, and maintenance requirements.

As mentioned above, the FSM integrated the optimization of fleet assignment and aircraft routing. Its objective is to select a set of augmented strings so as to minimize the total cost, while satisfying the relevant constraints. Each flight segment is assigned to only one fleet, and each flight can be assigned to exactly one *rotation* (routes that begin and end at the same station). A *count-time* was selected in order to count the aircraft on the ground and in the air. Note that all the events were sorted and numbered in increasing order of time.

Network Parameters:

S: set of augmented strings, indexed by $s \in S$.

K: set of fleets, indexed by $k \in K$.

F: set of flights, indexed by $i \in F$.

T: periodic time horizon over which the set of flight repeats.

$e^k_{i,a}/e^k_{i,d}$: event number for fleet k corresponding to the arrival/departure of flight i at some maintenance station.

$e^{+,k}_{i,a}/e^{+,k}_{i,d}$: next event number for fleet k at that station after the arrival/departure of flight i.

$e^{-,k}_{i,a}/e^{-,k}_{i,d}$: preceding event number for fleet k at that station before the arrival/departure of flight i.

$\big(y^k_{(e^{-,k}_{i,a}, e^k_{i,a})}$ and $y^k_{(e^k_{i,a}, e^{+,k}_{i,a})}\big)/\big(y^k_{(e^{-,k}_{i,d}, e^k_{i,d})}$ and $y^k_{(e^k_{i,d}, e^{+,k}_{i,d})}\big)$: ground arc variables that denote the number of aircraft of fleet k on the ground at that station between the predecessor event of flight i and the arrival/departure of i, and between the arrival/departure of i and its successor event, respectively.

G^k: set of ground arc variables for fleet k.

S^-_i/S^+_i: set of augmented strings ending/beginning with flight i.

a_{is}: binary indicator that equals 1 if flight $i \in F$ is in augmented string s, 0 otherwise.

c^k_s: cost of flying augmented string s with fleet k.

r^k_s: number of times augmented string s that is assigned to fleet k crosses the count-time.

p^k_j: number of times ground arc $j \in G^k$ for fleet k crosses the count-time.

N^k: number of aircraft in fleet k.

Note that r^k_s can be any nonnegative integer since a string may be longer than T time units, but p^k_j can only take on the values of 0 or 1 because ground arcs are at most T time units long, by definition.

Decision Variables:

x^k_s: augmented string variable that equals 1 if $s \in S$ is flown by fleet k, and 0 otherwise.

y^k_j: ground arc variable that represent the number of aircraft of fleet k on the ground arc j.

$$\textbf{FSM:} \quad \text{Minimize} \quad \sum_{k \in K} \sum_{s \in S} c_s^k x_s^k \tag{2.98}$$

subject to:

$$\sum_{k \in K} \sum_{s \in S} a_{is} x_s^k = 1, \qquad \forall i \in F, \tag{2.99}$$

$$\sum_{s \in S_i^+} x_s^k - y^k_{\left(e^{-,k}_{i,d}, e^k_{i,d}\right)} + y^k_{\left(e^k_{i,d}, e^{+,k}_{i,d}\right)} = 0, \qquad \forall i \in F, k \in K, \tag{2.100}$$

$$-\sum_{s \in S_i^-} x_s^k - y^k_{\left(e^{-,k}_{i,a}, e^k_{i,a}\right)} + y^k_{\left(e^k_{i,a}, e^{+,k}_{i,a}\right)} = 0, \qquad \forall i \in F, k \in K, \tag{2.101}$$

$$\sum_{s \in S} r_s^k x_s^k + \sum_{j \in G^k} p_j^k y_j^k \leq N^k, \qquad \forall k \in K, \tag{2.102}$$

$$y_j^k \geq 0, \ \text{integer}, \qquad \forall j \in G^k, k \in K, \tag{2.103}$$

$$x_s^k \in \{0, 1\}, \qquad \forall s \in S, k \in K. \tag{2.104}$$

The objective function is to minimize the total cost of selected strings. Constraint (2.99) requires that each flight is covered by exactly one string. Constraints (2.100) and (2.101) ensure the balance of flow, i.e., they equate the number of aircraft of fleet k arriving at and departing from a station. Constraint (2.102) asserts that the total number of aircraft in use should be no greater than the available fleet size. The last two constraints (2.103) and (2.104) represent logical restrictions. Although the model contains a large number of variables, practical-sized problems were demonstrated to be solvable to optimality using a branch-and-price technique.

In order to maintain even wear-and-tear in the long term, airlines require that every aircraft in a fleet flies all the flights assigned to its fleet, especially in short-haul operations. Since the hub-and-spoke structure accommodates a more flexible short-haul flying pattern, and it also increases the number of possible rotation permutations, the authors argued that the sequential approach of fleeting and routing does not have a significant detrimental effect. Therefore, Barnhart et al. (1998a) focused only on the routing subproblem and assumed that the fleeting has been determined *a priori*. Also, the schedule was assumed to repeat daily.

To assign the same sequence of flights to every single aircraft in the fleet, the proposed model incorporated the following *connectivity constraints*:

- Each aircraft starts by serving a flight at a different point in the sequence, where the sequence is a cycle, that includes every flight leg assigned to the fleet.
- The fleet's daily flight assignment is partitioned within the fleet, and the partitions are ordered so that they can be flown in turn.
- After the number of days same as the fleet size, each aircraft has flown all the assigned flights exactly once.

The authors proposed subtour elimination constraints as in the asymmetric traveling salesman problem (ATSP) to enforce connectivity.

Additional Parameters:

M: set of maintenance arcs.

M_i^+/M_i^-: set of maintenance connection arcs leaving/coming into flight i.

C: set of connection arcs for a specific fleet.

C_M: set of maintenance connection arcs.

$\hat{F} \subseteq F$: subset of F such that $2 \leq |\hat{F}| \leq \left\lfloor \frac{|F|}{2} \right\rfloor$.

$\delta^+(\hat{F}) = \{(i,j) \in C : i \in \hat{F}, j \in F \setminus \hat{F}\}$.

$\eta_{\hat{F}}^s$: binary indicator that equals 1 if string s leaves the set $\hat{F}$, and 0 otherwise.

Additional variables:

z_j: binary variable that equals 1 if the maintenance connection arc $j \in C_M$ is used, and 0 otherwise.

The proposed equal utilization routing (EUR) model can thus be formulated as follows:

$$\textbf{EUR:} \quad \text{Minimize} \quad \sum_{s \in S} c_s x_s \tag{2.105}$$

subject to:

$$\sum_{k\in K}\sum_{s\in S} a_{is}x_s = 1, \qquad \forall i \in F, \tag{2.106}$$

$$\sum_{s\in S_i^+} x_s = \sum_{j\in M_i^+} z_j, \qquad \forall i \in F, \tag{2.107}$$

$$\sum_{s\in S_i^-} x_s = \sum_{j\in M_i^-} z_j, \qquad \forall i \in F, \tag{2.108}$$

$$\sum_{s\in S} r_s x_s + \sum_{j\in M} p_j z_j \leq N, \tag{2.109}$$

$$\sum_{s\in S} \eta_{\hat{F}}^s x_s + \sum_{j\in C_M\cap\delta^+(\hat{F})} z_j \geq 1, \quad \forall \hat{F} \subseteq F, 2 \leq |\hat{F}| \leq \left\lfloor \frac{|F|}{2} \right\rfloor, \tag{2.110}$$

$$z_j \in \{0,1\}, \qquad \forall j \in C_M, \tag{2.111}$$

$$x_s \in \{0,1\}, \qquad \forall s \in S. \tag{2.112}$$

In this model, the superscript k was eliminated since the formulation only considered a single fleet type. The objective function (2.105) and Constraints (2.106)-(2.109), (2.111), and (2.112) are similar to those in the previous model, while Constraint (2.110) requires that the rotation is a cycle through the entire set of flights by enforcing that at least one string or at least one maintenance connection arc leaves each subset of flights. Note that this enforcement is not identical to the subtour elimination constraints in the traditional ATSP. Also, the connectivity constraints in this formulation were not determined *a priori* due to their scale; instead, they were added to the model on-the-fly whenever they were violated by solutions from the LP relaxations solved.

Haouari, Sherali, Mansour, and Aissaoui (2011b)

Haouari et al. (2011b) proposed a model for the aircraft fleeting and routing problem (AFRP) that combines fleet assignment with aircraft routing. Given a flight schedule in a study period (usually a week) and a fleet of different aircraft families, the problem is to determine a set of aircraft routes having a minimum total cost while meeting maintenance requirements. Specifically, the model minimized the total cost by constructing a feasible route (a sequence of flight legs) for each aircraft so as to cover each scheduled flight by a single aircraft, while satisfying the turn-time restrictions at each station along with the planned maintenance immobilizations.

Parameters:

F: set of available aircraft.

$F_f \subseteq F$: subset of aircraft belonging to aircraft type f ($f = 1, 2, \ldots, \phi$).

q_{ic}: number of seats for class c ($c = 1, 2, \ldots, C$) in aircraft type i, $i \in F_f$.

τ_i: turn-time for the aircraft type i, $i \in F_f$.

L: set of legs in the schedule.

L_i: set of flight legs that can be flown by aircraft i.

c_{ij}: flight cost when aircraft i is assigned to flight j.

s_{ijk}: deadhead flight cost when flight k's origin station differs from its preceding flight j's destination station, given that flights j and k are consecutively served by aircraft i.

$G_i = (N_i, A_i)$: time-space network, where N_i contains nodes that represent the flights $j \in L_i$ as well as a source node $o(i)$ and a sink node $d(i)$, and where A_i is the set of arcs that represent aircraft transitions, including arcs from source nodes and to sink nodes. Finally, define $N_i^* = N_i \setminus \{o(i), d(i)\}$.

Decision Variables:

y_{ij}: binary variable that takes the value 1 if aircraft i is assigned to flight j, and 0 otherwise.

x_{ijk}: binary variable that takes the value 1 if aircraft i is assigned to serve flight k immediately after flight j, and 0 otherwise.

This problem is formulated as follows:

$$\textbf{AFRP1:} \quad \text{Minimize} \quad \sum_{i\in F}\sum_{j\in N_i^*} c_{ij}y_{ij} + \sum_{i\in F}\sum_{(j,k)\in A_i} s_{ijk}x_{ijk} \tag{2.113}$$

subject to:

$$\sum_{k:\ (j,k)\in A_i} x_{ijk} = y_{ij}, \qquad \forall i \in F,\ j \in N_i \backslash \{d(i)\}, \tag{2.114}$$

$$\sum_{k:\ (k,j)\in A_i} x_{ikj} = y_{ij}, \qquad \forall i \in F,\ j \in N_i \backslash \{o(i)\}, \tag{2.115}$$

$$\sum_{i\in F:\ j\in N_i^*} y_{ij} = 1, \qquad \forall j \in L, \tag{2.116}$$

$$x_{ijk} \in \{0,1\}, \qquad \forall i \in F,\ (j,k) \in A_i, \tag{2.117}$$

$$y_{ij} \in \{0,1\}, \qquad \forall i \in F,\ j \in N_i,\ \text{with } y_{io(i)} = y_{id(i)}, \forall i \in F. \tag{2.118}$$

The objective function (2.113) minimizes the total cost incurred by fleet assignment. Constraints (2.114)-(2.115) require that each flight leg has exactly one predecessor and one successor, respectively. Constraint (2.116) assures that each flight is assigned to exactly one aircraft while satisfying planned maintenance immobilizations and compatibility constraints. The last two constraints (2.117) and (2.118) represent logical restrictions on the decision variables. This formulation has an assignment network structure for a fixed y feasible to (2.116) and (2.118); hence, Constraint (2.117) is relaxed to simply $x_{ijk} \geq 0, \forall i \in F, (j,k) \in A_i$.

Next, the authors also introduced a set partitioning based formulation, which is presented below.

Additional Parameters:

Ω^i: set of feasible routes including the dummy "ground route" $\big(o(i), d(i)\big)$ in the network $G_i, \forall i \in F$.

a_{ijr}: binary parameter that equals 1 if route $r \in \Omega^i$ covers flight $j \in L_i$, and 0 otherwise.

$\theta_{ijk} = c_{ij} + s_{ijk}$: cost of arc (j,k) in graph G_i.

$c_{ir} = \sum_{(j,k)\in r} \theta_{ijk}$: cost of assigning route $r \in \Omega^i$ to aircraft $i \in F$, including the assignment cost and the deadhead cost as appropriate.

Decision Variables:

z_{ir}: binary variable that takes the value 1 if route r is assigned to aircraft i, and 0 otherwise.

$$
\textbf{AFRP2:} \quad \text{Minimize} \quad \sum_{i\in F}\sum_{r\in\Omega^i} c_{ir} z_{ir} \tag{2.119}
$$

subject to:

$$
\sum_{i\in F}\sum_{r\in\Omega^i} a_{ijr} z_{ir} = 1, \qquad \forall j \in L, \tag{2.120}
$$

$$
\sum_{r\in\Omega^i} z_{ir} = 1, \qquad \forall i \in F, \tag{2.121}
$$

$$
z_{ir} \in \{0,1\}, \qquad \forall i \in F, r \in \Omega^i. \tag{2.122}
$$

Constraint (2.120) requires that each flight is covered exactly once by an aircraft that respects planned maintenance immobilizations and compatibility restrictions. Constraint (2.121) ensures that each aircraft is assigned to exactly one route. Constraint (2.122) states that the decision variables are binary-valued.

Models AFRP1 and AFRP2 can be extended to a profit-maximizing problem while considering itinerary- or path-based demands for each fare-class, by defining new integer variables ν_{pc} to denote the number of passengers for each fare-class c $(c = 1, 2, \ldots, C)$ that are accepted on each path $p \in P$.

Cohn and Barnhart (2003)

As mentioned before, when the set of short connections is determined during the stage of making aircraft maintenance routing decisions, it constrains the possible crew assignment since a crew group cannot change aircraft for two successive short-connected flights. Therefore, in order to account the impact on the subsequent crew pairing decisions, Cohn and Barnhart (2003) developed an extended crew pairing (ECP) modeling approach by delaying maintenance routing decisions and incorporating them partially into the crew pairing stage. In this paper, the authors focused on solution procedures that guarantee a maintenance-feasible crew pairing solution while considering a small number of maintenance routing decisions, where the methods achieved a flexible tradeoff between solution time and quality.

Cohn and Barnhart first introduced a basic integrated model (BIM) that combines the partition-based crew pairing model with the string-based maintenance routing model. However, two major drawbacks of this lightly hybrid model, namely, its large size and its weak LP relaxation, inhibited its use in real-scale problems.

Parameters:

F: set of flights.

R: set of feasible route strings.

R^T: set of route strings that span the count-time.

P: set of feasible pairings.

C: set of short connections.

N: set of activities in the time-space network.

N^T: set of nodes whose corresponding ground arcs span the count-time.

c_p: cost of pairing p.

δ_{fp}: binary indicator that equals 1 if pairing p contains flight f, and 0 otherwise.

α_{fr}: binary indicator that equals 1 if route string r contains flight f, and 0 otherwise.

ϑ_{cr}: binary indicator that equals 1 if route string r contains short connection c, and 0 otherwise.

η_{cp}: binary indicator variable that equals 1 if pairing p contains short connection c, and 0 otherwise.

K: size for a particular fleet.

Decision Variables:

d_r: binary decision variable that equals 1 if route string r is included in the solution, and 0 otherwise.

g_n^+/g_n^-: integer variable that equal the number of aircraft on the ground immediately following/preceding node n at a station.

y_p: binary decision variable that equals 1 if pairing p is included in the solution, and 0 otherwise.

$$
\begin{aligned}
\textbf{BIM:}\quad & \text{Minimize} && \sum_{p\in P} c_p y_p && && (2.123)\\
& \text{subject to:} && && && \\
& && \sum_{p\in P} \delta_{fp} y_p = 1, && \forall f\in F, && (2.124)\\
& && \sum_{r\in R} \alpha_{fr} d_r = 1, && \forall f\in F, && (2.125)\\
& && \sum_{r:\ \text{ends at } n} d_r + g_n^- - \sum_{r:\ \text{starts at } n} d_r - g_n^+ = 0, && \forall n\in N, && (2.126)\\
& && \sum_{r\in R^T} d_r + \sum_{n\in N^T} g_n^+ \le K, && && (2.127)\\
& && \sum_{r\in R} \vartheta_{cr} d_r - \sum_{p\in P} \eta_{cp} y_p \ge 0, && \forall c\in C, && (2.128)\\
& && d_r \in \{0,1\}, && \forall r\in R, && (2.129)\\
& && g_n^+,\ g_n^- \ge 0,\ \text{integer}, && \forall n\in N, && (2.130)\\
& && y_p \in \{0,1\}, && \forall p\in P. && (2.131)
\end{aligned}
$$

The objective function (2.123) minimizes the total cost for crew pairing. Constraints (2.124) and (2.125) require that each flight is covered by exactly one crew and one route string, respectively. Constraint (2.126) ensures the flow balance at each activity node, and Constraint (2.127) restricts the total number of aircraft used to its fleet size. Constraint (2.128) assures that a crew is not assigned to any short connection unless the same aircraft is assigned to it. This constraint plays a key role since it links maintenance routing decisions with crew pairings. Constraints (2.129)-(2.131) represent logical restrictions on the decision variables.

Since this aforementioned formulation has a large size and is weak in the LP sense, the authors proposed an *extended crew pairing* (ECP) model based on the traditional CP formulation.

The ECP model contains a convexity constraint that represents maintenance routing decisions, and that serves to select a single maintenance routing solution. The formulation of the ECP is presented below.

Additional parameters:

S: set of feasible maintenance routing decisions.

β_{cs}: binary indicator that equals 1 if maintenance routing s covers short connection c, and 0 otherwise.

Decision Variables:

x_s: binary decision variable that equals 1 if maintenance routing solution s is selected, and 0 otherwise.

$$\textbf{ECP:} \quad \text{Minimize} \quad \sum_{p \in P} c_p y_p \tag{2.132}$$

subject to:

$$\sum_{p \in P} \delta_{fp} y_p = 1, \qquad \forall f \in F, \tag{2.133}$$

$$\sum_{s \in S} \beta_{cs} x_s - \sum_{p \in P} \eta_{cp} y_p \geq 0, \qquad \forall c \in C, \tag{2.134}$$

$$\sum_{s \in S} x_s = 1, \tag{2.135}$$

$$x_s \in \{0, 1\}, \qquad \forall s \in S, \tag{2.136}$$

$$y_p \in \{0, 1\}, \qquad \forall p \in P. \tag{2.137}$$

The objective function (2.132) minimizes the total cost for pairings, as in the previous formulation. Constraint (2.133) requires that each flight is covered by exactly one crew pairing, and Constraint (2.134) ensures that a crew pairing is not assigned to a short connection unless it is covered in a selected maintenance routing solution. Constraint (2.135) guarantees that exactly one maintenance routing solution is selected. Constraints (2.136) and (2.137) place logical restrictions on the decision variables.

The authors further pointed out that only *unique and maximal* (UM) maintenance-feasible short connections need to be considered in the model. As is often the case, distinct feasible maintenance routes could involve the same set of short connections, which leads to an identical column representation in the ECP. Therefore, only one of these columns, referred to as *unique*, was actually incorporated. Moreover, a set of maintenance routing solutions is called *maximal* if it contains maintenance-feasible routes for which no additional short connections can be added while maintaining routing feasibility. The two techniques helped drastically reduce the number of candidate maintenance routing solutions, e.g., four out of 25,000 distinct solutions, as reported in their paper.

In order to identify the UM set, the authors proposed solving a series of maintenance routing problems with a different objective function and a side-constraint. Specifically, over the region defined by Constraints (2.125)-(2.127), (2.129)-(2.130), and using an additional constraint

$$\sum_{c \in C \setminus C^1} \sum_{r \in R} \vartheta_{cr} d_r \geq 1,$$

the model minimizes $\sum_{r \in R} -c_r d_r$, where c_r is defined as the number of short connections in route string r, and the set C^1 represents the set of short connections generated by the initial execution of the solution process. In this manner, a set of n UM maintenance-feasible routing solutions was generated by eliminating short connections from the previous set and repeating the run n times. To reduce the computational effort for subsequent runs, the solution obtained for any iteration was used to provide a warm-start solution for the next iteration.

The authors also proved that the ECP formulation has no more binary variables than the regular CP by showing that the residual maintenance model yields a polyhedron with integral vertices once the crew pairing variables are fixed. In addition, it was shown that the LP relaxation of the ECP is tighter than that of the BIM, which assures faster convergence.

Cordeau, Stojković, Soumis, and Desrosiers (2001)

Cordeau et al. (2001) integrated aircraft routing decisions within a crew pairing model and employed Benders decomposition to solve it. The authors assumed that the fleet assignment is given *a priori*, and the problem is to simultaneously determine a minimum-cost route for each aircraft along with crew pairings to serve the entire set of flight legs. The model only took into account the routine maintenance checks (A-checks), and ignores longer-period checks.

Parameters:

L: set of flight legs.

F: set of available aircraft.

K: set of available crews.

$G = (N, A)$: time-space network, where N is the set of nodes (representing flights), and A is the set of arcs (representing connections between flights).

o^f/d^f: origin/destination node of the aircraft of type f.

$O^f \subseteq \{(o^f, j)|j \in N\}$: set of arcs that link the origin node o^f of aircraft f to nodes (flights) that can be covered by this aircraft at the beginning of the horizon.

$D^f \subseteq \{(i, d^f)|i \in N\}$: set of arcs that link nodes (flights) that can be covered by aircraft f at the end of the horizon to the destination node d^f.

$G^f = (N^f, A^f)$: time-space network for aircraft f, where $N^f = N \cup \{o^f, d^f\}$, and $A^f = A \cup O^f \cup D^f$.

Note that o^k/d^k, O^k/D^k, and network $G^k = (N^k, A^k)$ are defined in a similar way for each crew pairing $k \in K$.

$C \subseteq A$: set of arcs that represent short connections in the network G.

Ω^f: set of feasible paths between node o^f and d^f in G^f for every aircraft $f \in F$.

Ω^k: set of feasible paths between node o^k and d^k in G^k for every crew pairing $k \in K$.

a^i_ω: binary indicator that equals 1 if node $i \in N^f$ belongs to path $\omega \in \Omega^f$, and 0 otherwise.

b^{ij}_ω: binary indicator that equals 1 if arc $(i, j) \in A^f$ belongs to path $\omega \in \Omega^f$, and 0 otherwise.

c_ω: cost of sending a unit of flow between o^f and d^f along path $\omega \in \Omega^f$.

Note that a^i_ω, b^{ij}_ω, and c_ω are defined similarly for every crew pairing path $\omega \in \Omega^k$.

Decision Variables:

θ_ω: binary decision variable that equals 1 if flow path ω for an aircraft is included in the solution, and 0 otherwise.

ζ_ω: binary decision variable that equals 1 if flow path ω for a crew pairing is included in the solution, and 0 otherwise.

The model formulation is then presented as follows:

$$\text{Minimize} \quad \sum_{f \in F} \sum_{\omega \in \Omega^f} c_\omega \theta_\omega + \sum_{k \in K} \sum_{\omega \in \Omega^k} c_\omega \zeta_\omega \tag{2.138}$$

subject to:

$$\sum_{f \in F} \sum_{\omega \in \Omega^f} a^i_\omega \theta_\omega = 1, \qquad \forall i \in N, \tag{2.139}$$

$$\sum_{k \in K} \sum_{\omega \in \Omega^k} a^i_\omega \zeta_\omega = 1, \qquad \forall i \in N, \tag{2.140}$$

$$\sum_{k \in K} \sum_{\omega \in \Omega^k} b^{ij}_\omega \zeta_\omega - \sum_{f \in F} \sum_{\omega \in \Omega^f} b^{ij}_\omega \theta_\omega \leq 0, \qquad \forall (i,j) \in C, \tag{2.141}$$

$$\sum_{\omega \in \Omega^f} \theta_\omega = 1, \qquad \forall f \in F, \tag{2.142}$$

$$\sum_{\omega \in \Omega^k} \zeta_\omega = 1, \qquad \forall k \in K, \tag{2.143}$$

$$\theta_\omega \in \{0,1\}, \qquad \forall \omega \in \Omega^f, \ f \in F, \tag{2.144}$$

$$\zeta_\omega \in \{0,1\}, \qquad \forall \omega \in \Omega^k, \ k \in K. \tag{2.145}$$

The objective function (2.138) minimizes the total cost for aircraft routing and crew scheduling decisions. Constraints (2.139) and (2.140) state that each flight leg is covered by exactly one aircraft and one crew pairing, respectively. Constraint (2.141) requires that a crew cannot possibly serve a short connection $(i,j) \in C$ unless some aircraft covers both flight i and flight j. Constraints (2.142) and (2.143) ensure that only one path is selected for each aircraft and each crew pairing, respectively. Constraints (2.144) and (2.145) represent logical restrictions on the decision variables.

Apart from the associated cost for aircraft and crews assignments, there exists another cost for repositioning them, i.e., the *deadhead* cost. The model can be modified to accommodate deadhead flights by splitting each node into two nodes linked by a regular flight arc and a deadhead arc.

It was observed that the computational time roughly grows quadratically with the number of constraints. Therefore, Benders decomposition was used to accelerate the solution process by decomposing the original model into a Benders master program and a series of subproblems. Each of these problems was solved by column generation, which was embedded in a branch-and-bound framework. A depth-first (diving) search was employed to find heuristic solutions through a process that sequentially rounds up a fractional decision variable whose value exceeds a particular threshold. Although possible, the authors did not report the need for backtracking whenever encountering infeasibility.

Following the presented ideas, a three-phase approach was developed to solve the overall problem. In Phase I, the LP relaxation of the proposed model was solved to optimality using Benders decomposition. Having the cuts thus generated, Phase II then reintroduced the integrality constraints for the aircraft routing variables and generated additional cuts while solving the master program and subproblems. Note that the solution process could stop at a suboptimal solution because of the heuristic branching performed on the aircraft path variables. In Phase III, integrality constraints on the crew path variables were added to the subproblems, and heuristic branching was likewise applied to the crew path variables. In the computational results, the authors reported solutions of very good quality obtained from this approach.

Mercier, Cordeau, and Soumis (2005)

Mercier et al. (2005) investigated an integrated model of aircraft routing and crew scheduling by further extending the model proposed by Cordeau et al. (2001) while incorporating additional features. They derived a tighter model representation and introduced several improvements in the solution approaches. The authors also benchmarked their results with respect to those of Cohn and Barnhart (2003), exhibiting improved performance.

The authors assumed that the fleet assignment is given *a priori*, and that crews can only serve on a particular fleet type; therefore, the integrated problem considers a particular fleet type. The objective function accounts for through-values of routing decisions and crew costs of pairings; moreover, it incorporates a penalty for restricted connections, where the latter event occurs when there is only a limited time to change aircraft for the next duty. This concept differs from that of short connections in that a restricted connection actually satisfies the minimum sit-time requirement, but is shorter than a preset ideal sit-time. Restricted connections are important to consider since they can potentially impair the robustness of the entire schedule for small unexpected perturbations.

The integrated model was formulated using a time-space network for both aircraft and crews. Besides flight arcs, connection arcs were added to the aircraft routing network in order to represent possible flight connections. At each station, within a 24-hour horizon, a connection arc links an arrival flight to any succeeding departure flight that satisfies the minimum plane turn-time restriction. In addition, the model also accommodated day-time maintenance requirements by discretizing a day into equal-length subperiods (e.g., two hours) and designating the starting period for each aircraft path. On the other hand, the crew network was modeled to contain arcs that represent regular flights, connections, and deadhead flights used to transport crews and aircraft. In contrast with aircraft routes, the crew pairings are acyclic. The proposed generalized integrated model is presented below.

Parameters:

$G^F = (N^F, A^F)$: time-space aircraft network, where N^F is the node set and A^F is the arc set.

$G^K = (N^K, A^K)$: time-space network of crews, where N^K is the node set and A^K is the arc set.

T: set of possible starting times for paths.

B: set of crew bases.

M: set of maintenance stations.

R: set of pairs of flight legs for which the connection between them is restricted.

S: set of pairs of flight legs for which the connection between them is short.

s^F_{mt}: source node that is linked to beginning flights from station m during time period t.

q^F_{mt}: sink node that is linked to ending flights at station m during time period t.

s^K_b: source node that represents the beginning of a pairing from station b.

q^K_b: sink node that represents the end of a pairing at base b.

$G^F_{mt} = (N^F_{mt}, A^F_{mt})$: aircraft network for starting station m, where $N^F_{mt} = N^F \cup \{s^F_{mt}\} \cup \{q^F_{mt}, \forall m \in M, t \in T\}$.

$G^K_b = (N^K_b, A^K_b)$: crew network for starting base b, where $N^K_b = N^K \cup \{s^K_b, q^K_b\}$.

Ω^{mt}: set of feasible paths from source node s^F_{mt} to a sink node in G^F_{mt}.

Ω^b: set of feasible paths from source node s^K_b to sink node $q^K_b \in G^K_b$.

a^i_ω: binary indicator that equals 1 if node i belongs to path ω, and 0 otherwise.

b^l_ω: binary indicator that equals 1 if flight leg l belongs to path ω, and 0 otherwise.

c_ω: cost of sending a unit of flow along path ω.

f_ω: number of aircraft required to cover path ω.

u_ω: number of duties in path ω.

p_{ij}: penalty cost associated with restricted connection $(l_i, l_j) \in R$.

p: length of the subperiod.

h: time required to perform a regular maintenance check (taken as a multiple of p).

n^{ij}_ω: binary indicator that equals 1 if flight leg i and j are performed in sequence in path ω, and 0 otherwise.

s_ω: number of short connections used by path ω.

ζ^F: number of available aircraft for this fleet type.

ζ^S: number of short connections allowed.

ζ^D: total number of duties allowed in all crew pairings.

Decision Variables:

χ_ω: binary variable that equals 1 if there is flow on crew path ω, and 0 otherwise.

θ_ω: binary variable that equals 1 if there is flow on aircraft path ω, and 0 otherwise.

R_{ij}: binary variable that equals 1 if restricted connection $(l_i, l_j) \in R$ is used by a crew but not by an aircraft, and 0 otherwise.

The generalized model for integrated aircraft routing and crew scheduling (GIM) can be stated as follows:

GIM: Minimize
$$\sum_{b \in B} \sum_{\omega \in \Omega^b} c_\omega \chi_\omega + \sum_{m \in M} \sum_{t \in T} \sum_{\omega \in \Omega^{mt}} c_\omega \theta_\omega + \sum_{(l_i, l_j) \in R} p_{ij} R_{ij} \tag{2.146}$$

subject to:
$$\sum_{m \in M} \sum_{t \in T} \sum_{\omega \in \Omega^{mt}} b^l_\omega \theta_\omega = 1, \qquad \forall l \in L, \tag{2.147}$$
$$\sum_{b \in B} \sum_{\omega \in \Omega^b} b^l_\omega \chi_\omega = 1, \qquad \forall l \in L, \tag{2.148}$$
$$\sum_{m \in M} \sum_{t \in T} \sum_{\omega \in \Omega^{mt}} f_\omega \theta_\omega \leq \zeta^F, \tag{2.149}$$
$$\sum_{m \in M} \sum_{t \in T} \sum_{\omega \in \Omega^{mt}} s_\omega \theta_\omega \leq \zeta^S, \tag{2.150}$$
$$\sum_{b \in B} \sum_{\omega \in \Omega^b} u_\omega \chi_\omega \leq \zeta^D, \tag{2.151}$$
$$\sum_{\omega \in \Omega^{mt}} a_\omega^{q^F_{mt}} \theta_\omega - \sum_{\omega \in \Omega^{mt}} a_\omega^{s^F_{m,t+1+\frac{h}{p}}} \theta_\omega = 0, \qquad \forall m \in M, t \in T, \tag{2.152}$$
$$\sum_{b \in B} \sum_{\omega \in \Omega^b} n^{ij}_\omega \chi_\omega - \sum_{m \in M} \sum_{t \in T} \sum_{\omega \in \Omega^{mt}} n^{ij}_\omega \theta_\omega \leq 0, \qquad \forall (l_i, l_j) \in S, \tag{2.153}$$
$$\sum_{b \in B} \sum_{\omega \in \Omega^b} n^{ij}_\omega \chi_\omega - \sum_{m \in M} \sum_{t \in T} \sum_{\omega \in \Omega^{mt}} n^{ij}_\omega \theta_\omega - R_{ij} \leq 0, \qquad \forall (l_i, l_j) \in R, \tag{2.154}$$
$$R_{ij} \in \{0,\ 1\}, \qquad \forall\, (l_i, l_j) \in R, \tag{2.155}$$
$$\theta_\omega \in \{0,\ 1\}, \qquad \forall m \in M, t \in T, \omega \in \Omega^{mt}, \tag{2.156}$$
$$\chi_\omega \in \{0,\ 1\}, \qquad \forall b \in B, \omega \in \Omega^b. \tag{2.157}$$

The objective function (2.146) minimizes the total cost of crew scheduling, aircraft routing, and penalties for restricted connections. Constraints (2.147) and (2.148) ensure that each flight leg is covered by exactly one aircraft and one crew pairing. Constraint (2.149) limits the total number of aircraft in service. Constraint (2.150) enforces that the total number of short connections cannot exceed a preset threshold for the purpose of a robust schedule. Constraint (2.151) restricts the total number of duties in the pairing schedule. Constraint (2.152) links the ending station of an aircraft path back to its starting station in order to maintain cyclic routes. Constraint (2.153) requires that short connections cannot be served by the same crew unless the aircraft also does not change. Constraint (2.154) imposes a penalty via the objective function for each restricted connection where crews change the assigned aircraft. Constraints (2.155)-(2.157) represent logical restrictions on the decision variables.

The model was solved by applying Benders decomposition. Instead of taking the aircraft routing problem (ARP) as the master program as in the natural order of operations, the authors proposed to use the crew pairing problem (CP) to formulate the Benders master program and the ARP as the subproblem. The subproblem mainly contains feasibility information and relegates the optimization decisions to the master program. It is worth noting that the ARP need not be solved to optimality since the subproblem does not account for much cost anyway.

As in Cordeau et al. (2001), a three-phase heuristic algorithm was applied to solve the problem. Phase I relaxed all the integrality constraints and solved the LP relaxation to optimality using Benders decomposition and column generation. Phase II recaptured the integrality constraints for the master problem, generated additional cuts, and reiterated. In Phase III, the integrality restrictions on the variables of the subproblem were reintroduced, and the resulting problem was solved once with fixed master program decisions.

Besides, a new heuristic branching strategy was also proposed to enhance algorithmic efficiency. In this procedure, within a depth-only search scheme, binary variables that exceed a given threshold λ_1 were all set to one. On the other hand, the binary variables whose fractional values lie between a smaller threshold λ_2 and λ_1 were sequentially set to one within the model. If neither case applied, then a branching on arcs or follow-ons was performed in a shortest-path network. This strategy accelerated the search to a near-optimal solution. Furthermore, to speed up the process, strong cuts were generated whenever the primal problem attained degenerate solutions. Specifically, using the ARP as the subproblem, a dominant (Pareto-optimal) cut was identified by taking values close to one as the core point and subsequently solving an auxiliary program.

Mercier and Soumis (2007)

In follow-on research, Mercier and Soumis (2007) improved their model that combines the aircraft routing problem and the crew scheduling problem in a single formulation. The crew cost is one of the major expenditures in an airline company and crew restrictions involve strict limits on the total number of landings, total work time, and total flight time. It has been variously shown that integrating aircraft routing with crew pairing can yield significantly better results than that obtained by solving the problems sequentially. In this paper, the authors further considered variable departure times for each leg, i.e., they permitted the departure times to slightly deviate from those in the original schedule. As demonstrated by their results, carefully chosen departure times can beneficially impact both aircraft routings and crew pairings.

The fleet assignment was assumed to be known *a priori*, and so the integrated problem decomposed into one for each fleet type. Given a set of daily flights for a specific aircraft type, the formulated model determines a minimum-cost set of aircraft routes and crew pairings with a modified schedule so as to cover each of the flights by exactly one aircraft and one crew.

Parameters:

L: set of flight legs.

U_i: set of possible departure times for flight leg i.

S: set of pairs of flight legs between which the connection time is short for at least one schedule combination.

S_{ij}: set of pairs of departure times $p \in U_i$ and $q \in U_j$ for which the connection time between flight legs i and j is short.

Ω^F: set of feasible aircraft paths.

Ω^K: set of feasible crew paths.

b_ω^i: binary indicator that equals 1 if flight leg i belongs to path ω, and 0 otherwise.

b_ω^{iu}: binary indicator that equals 1 if flight leg i with schedule u belongs to path ω, and 0 otherwise.

c_ω: cost of sending a unit of flow along path ω.

d_ω^{iu}: binary indicator that equals 1 if deadhead i with schedule u belongs to path ω, and 0 otherwise.

e_ω: number of duties in crew path ω.

f_ω: number of aircraft required to cover aircraft path ω.

n_ω^{ij}: binary indicator that equals 1 if flight legs i and j are served in sequence in path ω.

n_ω^{ijpq}: binary indicator that equals 1 if flight leg i with schedule p and flight leg j with schedule q are served in sequence in path ω.

ζ^F: number of available aircraft.

ζ^D: total number of duties allowed in all crew pairings.

Decision Variables:

χ_ω: binary variable representing the flow on crew path ω.

θ_ω: binary variable representing the flow on aircraft path ω.

This model is formulated as follows:

$$\text{Minimize} \quad \sum_{\omega \in \Omega^K} c_\omega \chi_\omega + \sum_{\omega \in \Omega^F} c_\omega \theta_\omega \tag{2.158}$$

subject to:

$$\sum_{\omega \in \Omega^F} b_\omega^i \theta_\omega = 1, \qquad \forall i \in L, \tag{2.159}$$

$$\sum_{\omega \in \Omega^F} f_\omega \theta_\omega \leq \zeta^F, \tag{2.160}$$

$$\sum_{\omega \in \Omega^K} b_\omega^i \chi_\omega = 1, \qquad \forall i \in L, \tag{2.161}$$

$$\sum_{\omega \in \Omega^K} e_\omega \chi_\omega \leq \zeta^D, \tag{2.162}$$

$$\sum_{\omega \in \Omega^K} d_\omega^{iu} \chi_\omega - \sum_{\omega \in \Omega^K} b_\omega^{iu} \chi_\omega \leq 0, \qquad \forall u \in U_i, i \in L, \tag{2.163}$$

$$\sum_{\omega \in \Omega^K} b_\omega^{iu} \chi_\omega - \sum_{\omega \in \Omega^F} b_\omega^{iu} \theta_\omega = 0, \qquad \forall u \in U_i, i \in L, \tag{2.164}$$

$$\sum_{\omega \in \Omega^K} n_\omega^{ijpq} \chi_\omega - \sum_{\omega \in \Omega^F} n_\omega^{ijpq} \theta_\omega \leq 0, \qquad \forall (p,q) \in S_{ij}, (i,j) \in S, \tag{2.165}$$

$$\theta_\omega \in \{0,1\}, \qquad \omega \in \Omega^F, \tag{2.166}$$

$$\chi_\omega \in \{0,1\}, \qquad \omega \in \Omega^K. \tag{2.167}$$

The objective function (2.158) is to minimize the total cost of crew pairing and aircraft routing. Note that since a large part of crew costs are already fixed by the predefined fleet assignment, the model only considers *through-values* (negative costs) and those costs that can be reduced by improved planning. Constraints (2.159) and (2.161) ensure that each flight leg is assigned to an aircraft and a crew, respectively. Constraints (2.160) and (2.162) place limits on the number of available aircraft and the total number of duties, respectively. Constraint (2.163) enforces the same schedule to be selected for the working crew and for the traveling crew (deadhead). Constraints (2.164) and (2.165) relate aircraft routing variables to crew variables, i.e., (2.164) requires that the aircraft and the crew choose the same schedule for every flight leg, and (2.165) assures that a crew does not change aircraft if the connection time is short. Finally, Constraints (2.166) and (2.167) are logical restrictions on the decision variables.

A simpler formulation was also proposed for the problem in which the number of constraints in (2.165) was reduced by considering an aggregated constraint as follows:

$$\sum_{\omega \in \Omega^K} n_\omega^{ij} \chi_\omega - \sum_{\omega \in \Omega^F} n_\omega^{ij} \theta_\omega \leq 0, \qquad \forall (i,j) \in S. \tag{2.168}$$

However, when using the aggregated restrictions (2.168) to replace (2.165), a larger integrality gap, or a greater number of fractional variables, was found to occur in the solution to the LP relaxation. Also, possible violations to the disaggregated constraints in (2.165) might exist within the resulting solution.

The number of variables in this path-based formulation grew exponentially with the number of nodes in the network. Furthermore, for each connection, several constraints were generated due to a set of possible departing/arriving times. This significantly adds to the model's complexity.

To summarize, the existing models in the literature address a variety of features of the airline scheduling process. The models for aircraft routing often require approaches based on decomposition techniques, e.g., column generation and Benders decomposition, in order to implicitly address restrictions for aircraft routing via sophisticated model implementations. In contrast, we present in the next chapter a *compact model* that explicitly incorporates the routing constraints within a node-arc formulation of polynomial size, which admits a relatively easy solution using off-the-shelf commercial solvers. A reformulation along with the derivation of valid inequalities is also performed to accelerate the solution process. We next propose in Chapter 4 an integrated model for fleeting, routing, and crew pairing decisions by imbedding the above aircraft routing model within this more general framework. The formulated integrated model is solved using tailored decomposition approaches along with several sophisticated acceleration techniques, which help deal effectively with the large-scale representation of the problem.

Chapter 3

A Lifted Compact Formulation for the Daily Aircraft Maintenance Routing Problem

3.1. Introduction

The principal airline operational problems include schedule planning, fleet assignment, aircraft routing, and crew scheduling. In this process, of particular interest in this present chapter, aircraft routing decisions are made after each flight leg has been assigned to an aircraft type. Specifically, the *aircraft routing problem* (ARP) determines the flying sequence, including periodic maintenance checks, for each individual aircraft in a given fleet. Although in the previous stage of fleet assignment, each flight leg is matched with a particular type of aircraft while optimizing profits, this does not take into consideration operational requirements for each individual aircraft. Consequently, the resulting solution therefore does not necessarily guarantee the existence of aircraft rotations that satisfy requirements such as the minimum turn-time at stations and fleet-specific maintenance check regulations, along with other restrictions on the total accumulated flying time, the total number of takeoffs, and the total number of days between two consecutive maintenance checks for a given aircraft. From this point of view, the ARP is significant to an airline company in that it generates operational feasible routes for aircraft while addressing the foregoing restrictions, perhaps iteratively by tweaking the previously fixed operational decisions as necessary.

This chapter focuses on deriving an aircraft maintenance routing scheme based on a set of daily repeated flight schedules; yet our model can be easily extended to multi-day periodic flight schedules. Specifically, we propose a new model that explicitly incorporates the routing and maintenance requirements within a node-arc formulation, which turns out to be polynomial in size. The initial representation contains some nonlinear features that are subsequently linearized by applying the Reformulation-Linearization Technique (RLT) of Sherali and Adams (1990, 1994). The RLT lifts the original model to an equivalent, tight, higher dimensional linear zero-one mixed-integer program while retaining its polynomial size aspect, which enables any suitable commercial solver (e.g., CPLEX) to obtain solutions quite readily without specialized algorithmic implementations.

The remainder of this chapter is organized as follows. In Section 3.2, we propose the new model formulation, first in its original form and then after being lifted by the RLT. A variety of realistic test instances (obtained from *United Airlines*) are solved in Section 3.4 to benchmark the performance of the proposed model. Finally, we present a summary and conclusions in Section 3.5.

3.2. Proposed Model Formulation

Instead of formulating the daily aircraft routing problem as a set partitioning problem and solving it using a branch-and-price approach, we model the problem using a novel polynomial-sized formulation that can be solved using a commercial solver (or, by a tailored branch-and-bound algorithm). In this section, we present a description of this proposed formulation.

Specifically, given a set of flight legs flown daily by a fleet of identical aircraft, and a set of stations where maintenance can be performed, the problem is to find a set of aircraft routes (or, rotations) such that:

- Each route is a sequence of flight legs to be flown consecutively by the same aircraft. The last flight leg should be compatible with the first flight leg, which thus yields a flight rotation.

- Each flight leg is covered by exactly one route.
- Each aircraft should visit maintenance stations (and spend at least the time duration required to perform a maintenance check) before: (i) accumulating a specified maximum flying time; (ii) accumulating a specified maximum number of takeoffs, and (iii) accumulating a specified maximum number of days.
- The total number of required aircraft should not exceed its specified fleet size.

The proposed model accounts for a flight schedule that repeats cyclically, a typical length of which can be a day or a week. In this dissertation, it is assumed that all the flights are designated within the time horizon from 0 to 1440 minutes (length of a day) recorded in the *Coordinated Universal Time* (UTC), and thus the pattern of flight arrivals and departures repeats cyclically on a daily basis. This proposed model also accounts for *wrap-around flights*, i.e., those flights that depart on a given day, cross the end of the time horizon, and arrive on the next day. Note that since the time definition in this model takes a more general form than those adopted in the fleet assignment model and/or other stages of the airline scheduling process, it actually relaxes the assumption that all the flights must be completed during the same day, thus permitting such wrap-around flights. However, if there exists a point in time over the 1440 minute block when no flights are in the air, then by defining $t = 0$ at such a “regeneration” point, we can redefine the data so that all flights are completed during the same day.

Furthermore, note that in this problem, we only consider the more frequently occurring mandated maintenance checks, i.e., the type A check, for each aircraft. Other aircraft maintenances are spaced over longer durations and, being more time-intensive, are therefore usually planned at a higher decision level whereby aircraft are appropriately periodically pulled out of and reinserted into service.

Notation:

Let L denote the set of flight legs assigned to the aircraft type under consideration. For each flight leg $j \in L$, we define the following notation (with all time durations expressed in minutes):

$DT_j \in [0, 1440]$: departure time of flight leg j.

$AT_j \in [0, 1440]$: arrival time of flight leg j.

DS_j: departure station of flight leg j.

AS_j: arrival station of flight leg j.

t_j: flying time of flight leg j (we assume that $t_j \leq 1440$ for practical reasons).

Also, we define:

τ: turn-time for the given aircraft type.

NA: number of available aircraft of the given type.

na: number of wrap-around flights in the schedule.

$t_{\max}$: maximum flying time between two consecutive maintenance checks (note that $t_j \leq t_{\max}, \forall j \in L$).

$to_{\max}$: maximum number of takeoffs between two consecutive maintenance checks ($to_{\max} \geq 1$).

$d_{\max}$: maximum number of days between two consecutive maintenance checks ($d_{\max} \geq 1$).

M: time duration of a maintenance check ($\tau < M < 1440$).

S: set of maintenance stations ($|S| \geq 1$).

For a given flight schedule, we define its associated digraph $G = (V, A)$ in which each node $j \in V$ represents a flight leg. We denote the origin/destination node of any arc $a \in A$ defined in the sequel by a^-/a^+. Also, we denote the set of arcs that are incident to/outgoing from node $j \in V$ by δ_j^-/δ_j^+, respectively. Moreover, each arc $a \in A$ represents a feasible connection, that is, an arc $a \in A$ if and only if an aircraft can consecutively serve flights a^- and a^+. More precisely, the set of arcs A is given by the union of six arc subsets A^1, A^2, A^3, A^4, A^5, and A^6 that are defined as follows:

- An arc $a \in A^1$ if and only if a maintenance check could be planned between the arrival of flight leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and both flight legs are required to be served consecutively on the same day. Hence, $(j, l) \in A^1 \Leftrightarrow$ (i)

$AS_j \equiv DS_l$; (ii) $AS_j \in S$; and (iii) $AT_j + M \leq DT_l$.

- An arc $a \in A^2$ if and only if a maintenance check could be planned between the arrival of flight leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and the same aircraft is required to serve flight leg l the day after serving flight leg j, even if $AT_j + \tau \leq DT_l$. Hence, $(j, l) \in A^2 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \in S$; and (iii) $DT_l < AT_j + M \leq DT_l + 1440$.

- An arc $a \in A^3$ if and only if a maintenance check could not be planned between the arrival of flight leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and both flight legs are required to be served consecutively on the same day. Hence, $(j, l) \in A^3 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \notin S$ or $DT_l < AT_j + M$; and (iii) $AT_j + \tau \leq DT_l$.

- An arc $a \in A^4$ if and only if a maintenance check could not be planned between the arrival of flight leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and the same aircraft serves flight leg l the day after serving flight leg j. Hence, $(j, l) \in A^4 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \notin S$ or $AT_j + M > DT_l + 1440$; and (iii) $DT_l < AT_j + \tau \leq DT_l + 1440$.

- An arc $a \in A^5$ if and only if a maintenance check could be planned between the arrival of leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and the same aircraft is required to serve flight leg l two days after serving flight leg j. This type of arc (see Figure 3.1) represents the situation in which the aircraft does not have enough time to undergo maintenance after serving flight j and then to serve flight l the next day. Hence, $(j, l) \in A^5 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \in S$; and (iii) $DT_l + 1440 < AT_j + M \leq DT_l + 2880$.

- An arc $a \in A^6$ if and only if a maintenance check could not be planned between the arrival of flight leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and the same aircraft is required to serve flight leg l two days after serving flight leg j (see Figure 3.2). Hence, $(j, l) \in A^6 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \notin S$; and (iii) $DT_l + 1440 < AT_j + \tau \leq DT_l + 2880$. Note that although the duration of this type of connection is more than 1440 minutes, the time spent by the aircraft between its arrival plus the turn-time and its departure is less than a day, i.e., $(DT_l + 2880) - AT_j - \tau < 1440$ by the first inequality in (iii).

Remark 3.1: It needs to be pointed out that the last two types of arcs, i.e., A^5 and A^6, account for special types of connections that seldom occur. Consider, for instance, a case where the arrival time for the only incoming flight to an airport is close to the end of a day

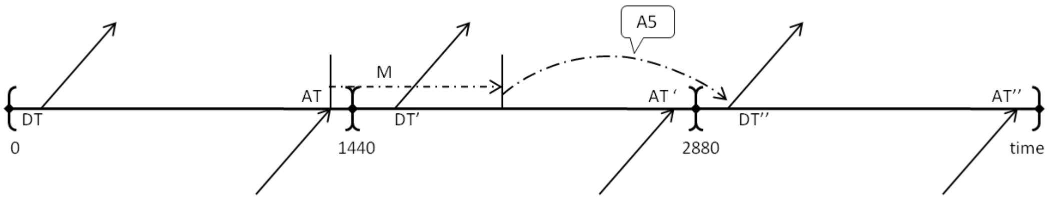

Figure 3.1: An Example for Arc Type A^5

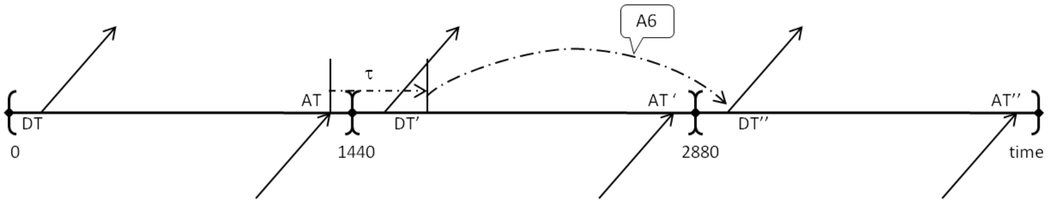

Figure 3.2: An Example for Arc Type A^6

and the departure of the only outgoing flight occurs so early in the day that the duration between the arrival and the next departure is less than the minimum turn-time (see the case of A^6 in Figure 3.2). Consequently, the aircraft has to wait another day on the ground. We therefore introduce these two types of arcs for the sake of the theoretical completeness of the model; however, typically in practice, for a regional set of flights, there exists a sufficiently long duration (exceeding M) beyond midnight or in the early morning hours when no activity occurs, so that the cyclical horizon [0, 1440] can be defined such that arcs of types A^5 and A^6 do not occur. □

Remark 3.2: It is worth noting that parallel arcs exist in G. Indeed, assume that we have two flight legs j and l satisfying: (i) $AS_j \equiv DS_l$; (ii) $AS_j \in S$; (iii) $DT_l < AT_j + M \leq DT_l + 1440$; and (iv) $AT_j + \tau \leq DT_l$. Then, from (i), (ii), and (iii), we have $(j, l) \in A^2$. Moreover, because of (i), (iii), and (iv), we have $(j, l) \in A^3$. The former case refers to the situation where the aircraft consecutively serves flight legs j and l on two consecutive days and is offered the opportunity to undergo a maintenance check between these two flight legs. The latter case corresponds to the situation where an aircraft consecutively serves flight legs j and l on the same day without being serviced in between. □

A *cycle* $(j_1, j_2, \ldots, j_p, j_1)$ in G corresponds to an aircraft rotation that consecutively covers flight legs $j_1, j_2, \ldots, j_p$ and back to j_1 in a cyclic fashion. If a cycle includes ξ_1 arcs belonging to $A^2 \cup A^4$ and ξ_2 arcs belonging to $A^5 \cup A^6$, and also covers ξ_3 wrap-around flights, then the corresponding rotation spans $\xi = \xi_1 + 2\xi_2 + \xi_3$ consecutive days, and since each flight leg is scheduled every day, ξ aircraft should be assigned to this rotation with each aircraft serving this same sequence over ξ days, but staggered so that the union of the flights served by each of these ξ aircraft each day equals $(j_1, \ldots, j_p)$ while their intersection is empty. Also, if a rotation includes an arc $a \in A^1 \cup A^2 \cup A^5$, then an aircraft assigned to this rotation has the opportunity to (but may not necessarily) undergo a maintenance check between flight legs a^- and a^+.

Consider, for example, a two-flight case between two stations A and B: Flight 1 traverses from A to B and spans the duration [5, 180], and Flight 2 traverses from B to A and spans the duration [1200, 1439], where Station A is a maintenance station. We further assume that the minimum turn-time is 30 minutes, while the maintenance time is 360 minutes. Note that in this simple case, the connection between Flight 1 and Flight 2 (at Station B) is of type A^3, but there does not exist enough time for Flight 2 to immediately connect to the next departure of Flight 1 at Station A. Hence, an aircraft serving Flight 2 on the first day would arrive at Station A toward the end of the first day (at $t = 1439$), and would then wait on the ground for a whole day (and possibly undergo maintenance) in order to connect to the next departing Flight 1 on the third day. Therefore, the connection between Flight 2 and Flight 1 (at Station A) is of arc type A^5. Consequently, the cycle including Flights 1 and 2 involves two aircraft ($\xi_1 = 0$, $\xi_2 = 1$, and $\xi_3 = 0$), each of which serves both the Flights 1 and 2 on alternate days. In this same example, note that if we were to change the duration of Flight 2 to [1200, 1], then the connection between Flight 2 and Flight 1 (at Station A) becomes of type A^2, but now we have $\xi_1 = 1$, $\xi_2 = 0$, and $\xi_3 = 1$, and so we again need two aircraft to serve this daily schedule in the same fashion as before.

The problem posed is to find a node partition such that each node is covered by exactly one maintenance-feasible rotation.

Decision Variables:

x_a: binary variable that equals 1 if arc $a \in A$ is selected, and 0 otherwise.

u_j: total accumulated flying hours for an aircraft since its last maintenance check after serving flight leg $j \in L$.

v_j: total number of takeoffs for an aircraft since its last maintenance check after serving flight leg $j \in L$.

d_j: total number of days for an aircraft since the last maintenance check after serving flight leg $j \in L$.

The problem can then be formulated as follows, where for convenience, we denote the set of maintenance permitting arcs as $A_M \equiv A^1 \cup A^2 \cup A^5$, and the set of non-maintenance arcs as $A_{NM} \equiv A^3 \cup A^4 \cup A^6$:

$$\textbf{ARP:} \quad \text{Find} \quad x, u, v, d \tag{3.1}$$

subject to:

$$\sum_{a \in \delta_j^-} x_a = 1, \qquad \forall j \in L, \tag{3.2}$$

$$\sum_{a \in \delta_j^+} x_a = 1, \qquad \forall j \in L, \tag{3.3}$$

$$u_j x_a = t_j x_a, \qquad \forall j \in L, a \in \delta_j^- \cap A_M, \tag{3.4}$$

$$u_j x_a = (u_{a^-} + t_j) x_a, \qquad \forall j \in L, a \in \delta_j^- \cap A_{NM}, \tag{3.5}$$

$$v_j x_a = x_a, \qquad \forall j \in L, a \in \delta_j^- \cap A_M, \tag{3.6}$$

$$v_j x_a = (v_{a^-} + 1) x_a, \qquad \forall j \in L, a \in \delta_j^- \cap A_{NM}, \tag{3.7}$$

$$d_j x_a = x_a, \qquad \forall j \in L, a \in \delta_j^- \cap A_M, \tag{3.8}$$

$$d_j x_a = d_{a^-} x_a, \qquad \forall j \in L, a \in \delta_j^- \cap A^3, \tag{3.9}$$

$$d_j x_a = (d_{a^-} + 1) x_a, \qquad \forall j \in L, a \in \delta_j^- \cap A^4, \tag{3.10}$$

$$d_j x_a = (d_{a^-} + 2) x_a, \qquad \forall j \in L, a \in \delta_j^- \cap A^6, \tag{3.11}$$

$$\sum_{a \in A^2 \cup A^4} x_a + 2 \sum_{a \in A^5 \cup A^6} x_a \leq NA - na, \tag{3.12}$$

$$t_j \leq u_j \leq t_{\max}, \qquad \forall j \in L, \tag{3.13}$$

$$1 \leq v_j \leq to_{\max}, \qquad \forall j \in L, \tag{3.14}$$

$$1 \leq d_j \leq d_{\max}, \qquad \forall j \in L, \tag{3.15}$$

$$x \text{ binary}. \tag{3.16}$$

Constraints (3.2)-(3.3) require that each flight leg has exactly one predecessor and one successor, respectively. Hence, together with (3.16), these restrictions induce the solution to be comprised of cycles or cyclic rotations. The nonlinear constraints (3.4)-(3.5) together with (3.13) enforce that the total flying time restriction is satisfied. Note that the nature of these constraints precludes a cyclic rotation with no maintenance visit. Similarly, Constraints (3.6)-(3.7) and (3.14) assure the restriction on the maximal number of takeoffs, and Constraints (3.8)-(3.11) and (3.15) guarantee the restriction on the maximum number of days between maintenance checks. Observe that, given binary values of x feasible to Constraints (3.2) and (3.3), Constraints (3.6)-(3.11) automatically induce the v- and d-variables to be integer-valued; hence, these variables are logically declared in (3.13) and (3.14) to be simply continuous-valued. (Likewise, if t_j are integer-valued, then so are the u-variable values in a feasible solution.) Constraint (3.12) requires that the total number of aircraft in service as accounted for each cyclic rotation above (where the sum of the ξ_3-values equals na) should not exceed the available size of fleet. Note that this is similar to an aircraft count constraint that is used in fleet assignment models (e.g., see Sherali et al. (2006)), where the accounting in this case is done at the count timeline $t = 1440$, and where the left-hand side in (3.12) counts the number of aircraft on the ground and na equals the number in the air at that time. Finally, Constraint (3.16) imposes logical restrictions on the binary decision variables.

Remark 3.3: Whereas the same daily schedule is considered here, the exact same model (with obvious changes) can be used for the same weekly schedule case (or for any other periodic repetition of flight schedules). □

Remark 3.4: Assume that the maximum daily number of aircraft that can visit a maintenance station $s \in S$ cannot exceed ν_s. Then, it is possible to accommodate this additional restriction by adding the upper-bounding constraint

$$\sum_{a \in A_s} x_a \leq \nu_s, \qquad \forall s \in S, \tag{3.17}$$

where $A_s \equiv \{a \in A_M : AS_{a^-} \equiv s\}, \forall s \in S$. It is noteworthy that, in practice, since a visit to a maintenance station does not necessarily translate into a maintenance check, the upper bound ν_s used in (3.17) can be taken as a suitable overestimate of the actual maintenance station capacity (say, by multiplying this capacity by a factor that is based on historical data of visits versus performed maintenance operations). In so doing, Constraint (3.17) limits the number of times opportunities are posed for each station (where each opportunity triggers a resetting of the time-clock in the model formulation). Such a constraint would thus avoid using a station having a low capacity from being overused with assumed-to-be-realized opportunities. □

3.2.1 Reformulation and Linearization

To enhance the solvability of Problem ARP, we next propose to apply the Reformulation-Linearization Technique (RLT) of Sherali and Adams (1990, 1994) to derive a tight, equivalent linear model representation of Problem ARP. To begin with, consider Constraints (3.4)-(3.5). We can linearize these constraints by using the substitution:

$$\omega_{jl} = u_j x_{jl}, \qquad \forall (j,l) \in A, \tag{3.18}$$

$$\rho_{jl} = u_l x_{jl}, \qquad \forall (j,l) \in A. \tag{3.19}$$

Thus, (3.4) and (3.5) get transformed to the following:

$$\rho_{jl} = t_l x_{jl}, \qquad \forall (j,l) \in A_M, \tag{3.20}$$

$$\rho_{jl} = \omega_{jl} + t_l x_{jl}, \qquad \forall (j,l) \in A_{NM}. \tag{3.21}$$

Also, multiplying (3.13) by each x_{lj} and x_{jl}, in δ_j^- and δ_j^+, respectively, and linearizing, we obtain (upon rearranging indices for the first case):

$$t_l x_{jl} \leq \rho_{jl} \leq t_{\max} x_{jl}, \forall (j,l) \in A, \tag{3.22}$$

and

$$t_j x_{jl} \leq \omega_{jl} \leq t_{\max} x_{jl}, \forall (j,l) \in A. \tag{3.23}$$

Furthermore, multiplying each of (3.2) and (3.3) by its corresponding u_j and linearizing, we get

$$\sum_{l:(l,j)\in A} \rho_{lj} = u_j, \qquad \forall j \in L, \tag{3.24}$$

$$\sum_{l:(j,l)\in A} \omega_{jl} = u_j, \qquad \forall j \in L. \tag{3.25}$$

Proposition 3.1. *For binary feasible x, Constraints (3.20)-(3.25) imply (3.4), (3.5), (3.13), (3.18), and (3.19), and so can replace these restrictions in Model ARP.*

Proof. The proof follows from Sherali et al. (1998) by noting that (3.20)-(3.25) are generated by taking suitable products of the restrictions in the specially structured set $S = \{x : (3.2), (3.3), x \geq 0\}$, which implies the bounds $0 \leq x_a \leq 1, \forall a \in A$, with the bounding restrictions (3.13) on the u-variables, and then linearizing the resulting constraints by using the substitution (3.18)-(3.19). □

We can further reduce the size of the reformulated problem (specifically addressing Constraints (3.20)-(3.25) for now) while retaining the strength of the underlying linear programming (LP) relaxation as established by the following result, where, with obvious notation, we use the vectors x, ω, u, and ρ to represent the corresponding subscripted variables (and similarly for other variables in the sequel):

Proposition 3.2. *Constraints (3.20)-(3.25) can be replaced by the following restrictions (3.26)-(3.28), which, together with (3.2), are equivalent to these constraints even in the continuous (LP relaxation) sense:*

$$\sum_{l:(j,l)\in A} \omega_{jl} = t_j + \sum_{l:(l,j)\in A_{NM}} \omega_{lj}, \qquad \forall j \in L, \tag{3.26}$$

$$t_j x_{jl} \leq \omega_{jl} \leq (t_{\max} - t_l) x_{jl}, \qquad \forall (j,l) \in A_{NM}, \tag{3.27}$$

$$t_j x_{jl} \leq \omega_{jl} \leq t_{\max} x_{jl}, \qquad \forall (j,l) \in A_M. \tag{3.28}$$

Proof. We can derive a set of constraints equivalent to (3.20)-(3.25) along with (3.2) (even in the continuous sense) as follows. First, we use the identities (3.20) and (3.21) to eliminate the ρ-variables from these restrictions. Accordingly, by substituting (3.20) and (3.21) into (3.24) and noting that $\sum_{l:(l,j)\in A} x_{lj} = 1$ by (3.2), we obtain

$$u_j = t_j + \sum_{l:(l,j)\in A_{NM}} \omega_{lj}, \qquad \forall j \in L.$$

We next eliminate the u-variables by combining the foregoing equation with (3.25), which yields (3.26). Next, observe that by substituting (3.20) into (3.22) we obtain a redundant relationship since $t_l \leq T_{\max}$. Furthermore, by substituting (3.21) into (3.22), and combining this with (3.23) for $(j,l) \in A_{NM}$ to eliminate implied bounding inequalities on the

ω-variables, we obtain (3.27), where we retain the inequalities in (3.23) for $(j, l) \in A_M$, as indicated in (3.28). Hence, along with (3.2), (x, ω) is feasible to (3.26)-(3.28) if and only if there exists a (u, ρ) such that (x, ω, u, ρ) is feasible to (3.20)-(3.25). □

Note that after solving the resulting model upon applying Proposition 3.2, the values of the u-variables that are present in the original Model ARP can be recovered, if so desired, by using the identities (3.25).

Similarly, for linearizing (3.6)-(3.11) in a likewise fashion, define

$$\eta_{jl} = v_j x_{jl}, \qquad \forall (j, l) \in A, \qquad (3.29)$$

$$\theta_{jl} = v_l x_{jl}, \qquad \forall (j, l) \in A; \qquad (3.30)$$

and

$$\lambda_{jl} = d_j x_{jl}, \qquad \forall (j, l) \in A, \qquad (3.31)$$

$$\mu_{jl} = d_l x_{jl}, \qquad \forall (j, l) \in A. \qquad (3.32)$$

Then, Constraints (3.6)-(3.7), and (3.8)-(3.11) can be rewritten as follows:

$$\theta_{jl} = x_{jl}, \qquad \forall (j, l) \in A_M, \qquad (3.33)$$

$$\theta_{jl} = \eta_{jl} + x_{jl}, \qquad \forall (j, l) \in A_{NM}, \qquad (3.34)$$

$$\mu_{jl} = x_{jl}, \qquad \forall (j, l) \in A_M, \qquad (3.35)$$

$$\mu_{jl} = \lambda_{jl}, \qquad \forall (j, l) \in A^3, \qquad (3.36)$$

$$\mu_{jl} = \lambda_{jl} + x_{jl}, \qquad \forall (j, l) \in A^4, \qquad (3.37)$$

$$\mu_{jl} = \lambda_{jl} + 2x_{jl}, \qquad \forall (j, l) \in A^6, \qquad (3.38)$$

where as above, from (3.14)-(3.15), we obtain

$$x_{jl} \leq \eta_{jl} \leq to_{\max} x_{jl}, \qquad \forall (j, l) \in A, \qquad (3.39)$$

$$x_{jl} \leq \theta_{jl} \leq to_{\max} x_{jl}, \qquad \forall (j, l) \in A, \qquad (3.40)$$

$$x_{jl} \leq \lambda_{jl} \leq d_{\max} x_{jl}, \qquad \forall (j, l) \in A, \qquad (3.41)$$

$$x_{jl} \leq \mu_{jl} \leq d_{\max} x_{jl}, \qquad \forall (j, l) \in A, \qquad (3.42)$$

and from Constraints (3.2) and (3.3), we obtain

$$\sum_{l:(l,j)\in A} \theta_{lj} = v_j, \qquad \forall j \in L, \tag{3.43}$$

$$\sum_{l:(j,l)\in A} \eta_{jl} = v_j, \qquad \forall j \in L, \tag{3.44}$$

$$\sum_{l:(l,j)\in A} \mu_{lj} = d_j, \qquad \forall j \in L, \tag{3.45}$$

$$\sum_{l:(j,l)\in A} \lambda_{jl} = d_j, \qquad \forall j \in L. \tag{3.46}$$

Following an argument similar to Proposition 3.2, we can eliminate the θ-, μ-, v-, and d-variables from (3.33)-(3.46), and effectively project these constraints onto the space of the (x, η, λ)-variables to obtain an equivalent set of relationships as follows:

$$\sum_{l:(j,l)\in A} \eta_{jl} = 1 + \sum_{l:(l,j)\in A_{NM}} \eta_{lj}, \qquad \forall j \in L, \tag{3.47}$$

$$x_{jl} \leq \eta_{jl} \leq (to_{\max} - 1)x_{jl}, \qquad \forall (j,l) \in A_{NM}, \tag{3.48}$$

$$x_{jl} \leq \eta_{jl} \leq to_{\max}x_{jl}, \qquad \forall (j,l) \in A_M, \tag{3.49}$$

$$\sum_{l:(j,l)\in A} \lambda_{jl} = \sum_{l:(l,j)\in A_M \cup A^4} x_{lj} + 2\sum_{l:(l,j)\in A^6} x_{lj} + \sum_{l:(l,j)\in A_{NM}} \lambda_{lj}, \qquad \forall j \in L, \tag{3.50}$$

$$x_{jl} \leq \lambda_{jl} \leq (d_{\max} - 1)x_{jl}, \qquad \forall (j,l) \in A^4, \tag{3.51}$$

$$x_{jl} \leq \lambda_{jl} \leq (d_{\max} - 2)x_{jl}, \qquad \forall (j,l) \in A^6, \tag{3.52}$$

$$x_{jl} \leq \lambda_{jl} \leq d_{\max}x_{jl}, \qquad \forall (j,l) \in A_M \cup A^3. \tag{3.53}$$

Accordingly, Problem ARP can be equivalently stated as the 0-1 mixed-integer program specified below, where for the sake of convenience, we define the following parameters:

$$b_a^t = \begin{cases} t_{\max} - t_{a^+}, \ \forall a \in A_{NM}, \\ t_{\max}, \ \forall a \in A_M, \end{cases}$$

$$b_a^{to} = \begin{cases} to_{\max} - 1, \ \forall a \in A_{NM}, \\ to_{\max}, \ \forall a \in A_M, \end{cases}$$

$$b_a^d = \begin{cases} d_{\max} - 1, \ \forall a \in A^4, \\ d_{\max} - 2, \ \forall a \in A^6, \\ d_{\max}, \ \forall a \in A_M \cup A^3. \end{cases}$$

ARP-RLT: Find $\quad x, \omega, \eta, \lambda \qquad (3.54)$

subject to:

$$\sum_{a \in \delta_j^-} x_a = 1, \qquad \forall j \in L, \quad (3.55)$$

$$\sum_{a \in \delta_j^+} x_a = 1, \qquad \forall j \in L, \quad (3.56)$$

$$\sum_{a \in \delta_j^+} \omega_a = t_j + \sum_{a \in \delta_j^- \cap A_{NM}} \omega_a, \qquad \forall j \in L, \quad (3.57)$$

$$t_a\text{-}x_a \leq \omega_a \leq b_a^t x_a, \qquad \forall a \in A, \quad (3.58)$$

$$\sum_{a \in \delta_j^+} \eta_a = 1 + \sum_{a \in \delta_j^- \cap A_{NM}} \eta_a, \qquad \forall j \in L, \quad (3.59)$$

$$x_a \leq \eta_a \leq b_a^{to} x_a, \qquad \forall a \in A, \quad (3.60)$$

$$\sum_{a \in \delta_j^+} \lambda_a = 1 + \sum_{a \in \delta_j^- \cap A^6} x_a - \sum_{a \in \delta_j^- \cap A^3} x_a + \sum_{a \in \delta_j^- \cap A_{NM}} \lambda_a, \quad \forall j \in L, \quad (3.61)$$

$$x_a \leq \lambda_a \leq b_a^d x_a, \qquad \forall a \in A, \quad (3.62)$$

$$\sum_{j \in L} \left(1 - \sum_{a \in \delta_j^- \cap (A^1 \cup A^3)} x_a + \sum_{a \in \delta_j^- \cap (A^5 \cup A^6)} x_a \right) \leq NA - na, \qquad (3.63)$$

$$x \text{ binary}, \qquad (3.64)$$

where we have used (3.55) to equivalently rewrite (3.50) and (3.12) as (3.61) and (3.63) above, respectively, in terms of more sparse constraints having unit coefficients for expediency in implementation. Also note that, as mentioned before (also see Proposition 3.1), only the x-variables are explicitly restricted to be integer-valued (binary), and all other variables are declared to be continuous, where similar to Model ARP, for given integer data, the resultant u-, v-, and d-variables will automatically take on integer values for any feasible solution to Problem ARP-RLT upon using the respective identities (3.25), (3.44), and (3.46).

3.3. Further Model Enhancements

In order to further tighten Model ARP-RLT, we first introduce a valid inequality in Section 3.3.1, and then propose two alternative techniques in Sections 3.3.2 and 3.3.3 for respectively augmenting the model either with certain partial convex hull representations or with a suitable set of valid inequalities implied thereby.

3.3.1 A Valid Inequality

Proposition 3.3. *Suppose that we lay out all the flights in the schedule as intervals on* $[0, 1440]$ *where the arrival time is extended by* τ *to account for the turn-time, and where the extended flight interval is wrapped-around if its end-point exceeds 1440. Let* γ_1 *denote the maximum number of such overlapping extended flight intervals at any point in time. Then the following inequality is valid:*

$$\sum_{a \in A^2 \cup A^4} x_a + 2 \sum_{a \in A^5 \cup A^6} x_a + na \geq \max\{\gamma_1, \gamma_2\}, \tag{3.65}$$

where $\gamma_2 = \left\lceil \frac{(M-\tau)\hat{\mu} + \sum_{j \in L}(t_j + \tau)}{1440} \right\rceil$*, and* $\hat{\mu} = \max\left\{ \left\lceil \frac{\sum_{j \in L} t_j}{t_{\max}} \right\rceil, \left\lceil \frac{|L|}{to_{\max}} \right\rceil \right\}$.

Proof: Note that if $\hat{N}$ represents the actual number of aircraft that are needed to serve the given flight schedule, then it must be true that $\hat{N} \geq \gamma_1$. Furthermore, denote by μ the (unknown) number of daily maintenance checks performed on average. Since the maximum flying time between two consecutive maintenance checks is $t_{\max}$, we have that $\mu \geq \lceil \frac{\sum_{j \in L} t_j}{t_{\max}} \rceil$. Also, since the maximum number of takeoffs between two consecutive maintenance checks is $to_{\max}$, this implies that $\mu \geq \lceil \frac{|L|}{to_{\max}} \rceil$. Hence, a valid lower bound on μ is given by $\hat{\mu}$ as stated in the proposition, and so a lower bound on the total average daily maintenance time is $M\hat{\mu}$. Moreover, $\sum_{j \in L}(t_j + \tau)$ represents the total aircraft-minutes of flying plus turn-times on a daily basis. Noting that the maintenance duration could account for the turn-time itself ($M > \tau$), we have that the total aircraft-minutes of activities accounted for on average over a day is at least $(M - \tau)\hat{\mu} + \sum_{j \in L}(t_j + \tau)$, whence $\hat{N} \geq \max\{\gamma_1, \gamma_2\}$. Noting from Constraint (3.12) that the left-hand side of Constraint (3.65) represents the total number of aircraft that are required to serve the particular routing scheme for any given feasible solution, this value must be at least as large as $\hat{N}$, which establishes the validity of (3.65). □

Observe that the value of γ_1 for use in (3.65) can be easily computed by counting the number of overlapping flights at the tail of each extended flight interval, for example, and selecting the maximum such value. Moreover, our computational experience reported in Section 3.4 shows that γ_1 yields, in general, a much tighter lower bound than γ_2, and always determined the right-hand side (RHS) in (3.65) for our test cases. In addition, we point out that the lower bound can be further tightened by solving the LP relaxation of the model with the objective function of minimizing the number of aircraft needed to serve the given flight schedule, and then including the rounded-up resulting value within the maximand on the RHS of (3.65). We comment here that although this yields a tighter inequality, we found that the additional preprocessing effort does not justify its use.

Henceforth, we will assume that Constraint (3.65) is incorporated within Model ARP-RLT as well as within its various augmentations discussed next.

3.3.2 Partial Convex Hull Representations (Model ARP-RLT$^+$)

To begin with, we first identify a special unimodular substructure that is inherent within Problem ARP-RLT, which will motivate the derivation of a further lifted model formulation based on constructing certain partial convex hull representations:

Proposition 3.4. *Define the set* $X \equiv \{(x, \omega, \eta, \lambda) : (3.55), (3.56), (3.58), (3.60), (3.62),\ \textit{and}\ x \geq 0\}$. *Then x is binary-valued at each extreme point of X.*

Proof: It is sufficient to show that for any arbitrary objective function to minimize $c_1^T x + c_2^T \omega + c_3^T \eta + c_4^T \lambda$, subject to $(x, \omega, \eta, \lambda) \in X$, there exists an optimal solution such that x is binary-valued. This follows since an optimal solution to the latter linear program can be obtained by first setting each component of ω, η, and λ at its respective lower or upper bounding function in (3.58), (3.60), and (3.62), according to whether the corresponding coefficient in the vectors c_2, c_3, and c_4 is nonnegative or negative, respectively. This equivalently reduces the given LP to the form of minimizing a resulting objective function $\tilde{c}^T x$ subject to (3.55), (3.56), and $x \geq 0$, which is a linear assignment problem and therefore has an optimum at which x is binary-valued. □

Proposition 3.4 asserts that any partial convexification process or derivation of additional valid inequalities for Model ARP-RLT must necessarily involve constraints from the set {(3.57), (3.59), (3.61), and (3.63)}. Accordingly, suppose that we solve the LP relaxation of ARP-RLT at the root node and obtain an optimal solution $(\bar{x}, \bar{\omega}, \bar{\eta}, \bar{\lambda})$. Naturally, if $\bar{x}$ is binary-valued, then this solution solves ARP-RLT. Otherwise, suppose that the set $J \equiv \{j \in L : (\bar{x}_a, a \in \delta_j^- \cup \delta_j^+)$ is not binary-valued$\}$ is nonempty. Given some selected algorithmic parameter $Q \in [1, |L|]$, we next select up to Q indices $j \in J$ in nonincreasing order of the total fractionality $\sum_{a \in \delta_j^- \cup \delta_j^+} \min\{\bar{x}_a, 1 - \bar{x}_a\}$ so long as the latter is positive, with ties broken in favor of higher values of the number of fractional values in $\{\bar{x}_a,\ a \in \delta_j^- \cup \delta_j^+\}$. Denote the set of indices thus selected as $J_Q \subseteq J$. For each $j \in J_Q$, consider the following corresponding subset of constraints from (3.55)-(3.62) and (3.64), rewritten for the sake of convenience:

$$\sum_{a \in \delta_j^-} x_a = 1, \tag{3.66}$$

$$\sum_{a \in \delta_j^+} x_a = 1, \tag{3.67}$$

$$\sum_{a \in \delta_j^+} \omega_a = t_j + \sum_{a \in \delta_j^- \cap A_{NM}} \omega_a, \tag{3.68}$$

$$\sum_{a \in \delta_j^+} \eta_a = 1 + \sum_{a \in \delta_j^- \cap A_{NM}} \eta_a, \tag{3.69}$$

$$\sum_{a \in \delta_j^+} \lambda_a = 1 + \sum_{a \in \delta_j^- \cap A^6} x_a - \sum_{a \in \delta_j^- \cap A^3} x_a + \sum_{a \in \delta_j^- \cap A_{NM}} \lambda_a, \tag{3.70}$$

$$t_a\text{-}x_a \le \omega_a \le b_a^t x_a, \qquad \forall a \in \delta_j^- \cup \delta_j^+, \tag{3.71}$$

$$x_a \le \eta_a \le b_a^{to} x_a, \qquad \forall a \in \delta_j^- \cup \delta_j^+, \tag{3.72}$$

$$x_a \le \lambda_a \le b_a^d x_a, \qquad \forall a \in \delta_j^- \cup \delta_j^+, \tag{3.73}$$

$$x_a \text{ binary}, \qquad \forall a \in \delta_j^- \cup \delta_j^+. \tag{3.74}$$

We will now apply the special-structured RLT approach of Sherali et al. (1998) (based on the special generalized upper bounding constraints (3.66) and (3.67)) in order to construct a lifting of (3.66)-(3.74) through a partial convexification process. Toward this end, first of all, note from (3.66) and (3.67), and the definitions of the variables ω, η, and λ as given respectively by (3.18), (3.29), and (3.31) that we have the following identities (where for the sake of clarity, we define $RS(j) \equiv \{l : (l, j) \in A\}$, and $FS(j) \equiv \{l : (j, l) \in A\}$ as the reverse and forward stars of j respectively):

$$x_{lj}x_{kj} = \omega_{lj}x_{kj} = \eta_{lj}x_{kj} = \lambda_{lj}x_{kj} = 0, \qquad \forall l, k \in RS(j), l \neq k; \tag{3.75}$$

$$x_{jl}x_{jk} = \omega_{jl}x_{jk} = \eta_{jl}x_{jk} = \lambda_{jl}x_{jk} = 0, \qquad \forall l, k \in FS(j), l \neq k; \tag{3.76}$$

$$\omega_a x_a = \omega_a, \eta_a x_a = \eta_a, \text{ and } \lambda_a x_a = \lambda_a, \qquad \forall a \in \delta_j^- \cup \delta_j^+. \tag{3.77}$$

Accordingly, we construct the following set of RLT product constraints (P$_1$)-(P$_5$):

P$_1$: Multiply (3.66) by each x_{jk}, ω_{jk}, η_{jk}, and λ_{jk}, $\forall k \in FS(j)$.

P$_2$: Multiply (3.67) by each x_{lj}, ω_{lj}, η_{lj}, and λ_{lj}, $\forall l \in RS(j)$.

P$_3$: Multiply each of (3.68)-(3.70) by each $x_{lj}, \forall l \in RS(j)$, and each $x_{jk}, \forall k \in FS(j)$.

P$_4$: Multiply each of (3.71)-(3.73) corresponding to $a \in \delta_j^-$ by each $x_a, \forall a \in \delta_j^+$, and corresponding to $a \in \delta_j^+$ by each $x_a, \forall a \in \delta_j^-$.

P$_5$: Impose the nonnegativity restrictions $x_{lj}x_{jk} \geq 0,\ \forall l \in RS(j),\ k \in FS(j)$.

We then linearize (P$_1$)-(P$_5$) by making the following substitutions:

$$\begin{array}{ll} y_{ljk} \equiv x_{lj}x_{jk}, & \forall l \in RS(j), k \in FS(j), \\ z^{\omega 1}_{ljk} \equiv \omega_{lj}x_{jk} \text{ and } z^{\omega 2}_{ljk} \equiv \omega_{jk}x_{lj}, & \forall l \in RS(j), k \in FS(j), \\ z^{\eta 1}_{ljk} \equiv \eta_{lj}x_{jk} \text{ and } z^{\eta 2}_{ljk} \equiv \eta_{jk}x_{lj}, & \forall l \in RS(j), k \in FS(j), \\ z^{\lambda 1}_{ljk} \equiv \lambda_{lj}x_{jk} \text{ and } z^{\lambda 2}_{ljk} \equiv \lambda_{jk}x_{lj}, & \forall l \in RS(j), k \in FS(j). \end{array} \tag{3.78}$$

For each $j \in J_Q$, let X^j denote the set of RLT constraints thus obtained via applying the substitution (3.78) along with the identities (3.75)-(3.77) to (P$_1$)-(P$_5$). We then add X^j to the formulation ARP-RLT and drop the consequently implied constraints (3.71)-(3.73) for each $j \in J_Q$. Let **ARP-RLT**$^+$ denote the resultant lifted model; as shown by Sherali et al. (1998), the lifting process for constructing ARP-RLT$^+$ incorporates the intersection of the convex hulls of (3.66)-(3.74) constructed with respect to enforcing binary restrictions on the variables in (3.66) and (3.67) separately for each $j \in J_Q$. Thus we call this a *partial convexification process*. To facilitate implementation, we provide below an explicit statement of the set X^j, for any $j \in J_Q$:

$$\sum_{l \in RS(j)} y_{ljk} = x_{jk}, \qquad \forall k \in FS(j), \tag{3.79}$$

$$\sum_{l \in RS(j)} z^{\omega 2}_{ljk} = \omega_{jk}, \qquad \forall k \in FS(j), \tag{3.80}$$

$$\sum_{l \in RS(j)} z^{\eta 2}_{ljk} = \eta_{jk}, \qquad \forall k \in FS(j), \tag{3.81}$$

$$\sum_{l \in RS(j)} z^{\lambda 2}_{ljk} = \lambda_{jk}, \qquad \forall k \in FS(j), \tag{3.82}$$

$$\sum_{k \in FS(j)} y_{ljk} = x_{lj}, \qquad \forall l \in RS(j), \tag{3.83}$$

$$\sum_{k \in FS(j)} z^{\omega 1}_{ljk} = \omega_{lj}, \qquad \forall l \in RS(j), \tag{3.84}$$

$$\sum_{k \in FS(j)} z^{\eta 1}_{ljk} = \eta_{lj}, \qquad \forall l \in RS(j), \tag{3.85}$$

$$\sum_{k \in FS(j)} z^{\lambda 1}_{ljk} = \lambda_{lj}, \qquad \forall l \in RS(j), \tag{3.86}$$

$$\sum_{k \in FS(j)} z^{\omega 2}_{ljk} = t_j x_{lj} + \{\omega_{lj} \text{ if } (l,j) \in A_{NM}; 0 \text{ otherwise}\}, \qquad \forall l \in RS(j), \tag{3.87}$$

$$\omega_{jk} = t_j x_{jk} + \sum_{l:(l,j) \in A_{NM}} z^{\omega 1}_{ljk}, \qquad \forall k \in FS(j), \tag{3.88}$$

$$\sum_{k \in FS(j)} z^{\eta 2}_{ljk} = x_{lj} + \{\eta_{lj} \text{ if } (l,j) \in A_{NM}; 0 \text{ otherwise}\}, \qquad \forall l \in RS(j), \tag{3.89}$$

$$\eta_{jk} = x_{jk} + \sum_{l:(l,j) \in A_{NM}} z^{\eta 1}_{ljk}, \qquad \forall k \in FS(j), \tag{3.90}$$

$$\begin{aligned} \sum_{k \in FS(j)} z^{\lambda 2}_{ljk} &= x_{lj} + \{\lambda_{lj} \text{ if } (l,j) \in A_{NM}; 0 \text{ otherwise}\} \\ &\quad + \{x_{lj} \text{ if } (l,j) \in A^6; -x_{lj} \text{ if } (l,j) \in A^3; 0 \text{ otherwise}\}, \qquad \forall l \in RS(j), \end{aligned} \tag{3.91}$$

$$\lambda_{jk} = x_{jk} + \sum_{l:(l,j)\in A^6} y_{ljk} - \sum_{l:(l,j)\in A^3} y_{ljk} + \sum_{l:(l,j)\in A_{NM}} z^{\lambda 1}_{ljk}, \qquad \forall k \in FS(j), \tag{3.92}$$

$$t_l y_{ljk} \leq z^{\omega 1}_{ljk} \leq b^t_{lj} y_{ljk}, \qquad \forall l \in RS(j), k \in FS(j), \tag{3.93}$$

$$t_j y_{ljk} \leq z^{\omega 2}_{ljk} \leq b^t_{jk} y_{ljk}, \qquad \forall l \in RS(j), k \in FS(j), \tag{3.94}$$

$$y_{ljk} \leq z^{\eta 1}_{ljk} \leq b^{to}_{lj} y_{ljk}, \qquad \forall l \in RS(j), k \in FS(j), \tag{3.95}$$

$$y_{ljk} \leq z^{\eta 2}_{ljk} \leq b^{to}_{jk} y_{ljk}, \qquad \forall l \in RS(j), k \in FS(j), \tag{3.96}$$

$$y_{ljk} \leq z^{\lambda 1}_{ljk} \leq b^{d}_{lj} y_{ljk}, \qquad \forall l \in RS(j), k \in FS(j), \tag{3.97}$$

$$y_{ljk} \leq z^{\lambda 2}_{ljk} \leq b^{d}_{jk} y_{ljk}, \qquad \forall l \in RS(j), k \in FS(j), \tag{3.98}$$

$$y_{ljk} \geq 0, \qquad \forall l \in RS(j), k \in FS(j). \tag{3.99}$$

Based on some preliminary experimental investigations, we adopted the following root node strategy that incorporates the foregoing RLT constraints:

Step 1: Solve the LP relaxation of Model ARP-RLT at the root node.

Step 2: For a selected parameter $1 \leq Q \leq |L|$, identify the index set J_Q, and generate the lifted RLT constraints $X^j, \forall j \in J_Q$, and add them to the model in order to construct ARP-RLT$^+$.

Step 3: Solve the resulting lifted Model ARP-RLT$^+$ as a mixed-integer program using an MIP solver (we used CPLEX for this purpose).

Note that optionally, we could repeat *Steps 1* and *2* after updating the model representation, but we found that performing this model augmentation just once gave promising results. Moreover, in order to curtail the size of the resulting formulation, we restricted $Q = 1$ in our computational experiments.

3.3.3 Alternative Technique for Using X^j (Model ARP-RLT*)

For any $j \in J_Q$, consider the set X^j given by Constraints (3.79)-(3.99), which can be rewritten compactly as follows for the sake of analytical convenience:

$$A\zeta^{2j} = B\zeta^{1j}, \tag{3.100}$$

$$D\zeta^{2j} \leq 0, \tag{3.101}$$

where ζ^{1j} represents the vector of the $(x, \omega, \eta, \lambda)$-variables defining the original set of constraints (3.66)-(3.74) from ARP-RLT, and where ζ^{2j} represents the new set of (y, $z^{\omega 1}$, $z^{\omega 2}$, $z^{\eta 1}$, $z^{\eta 2}$, $z^{\lambda 1}$, $z^{\lambda 2}$)-variables defined in (3.78). Furthermore, let $\bar{\zeta}^{1j}$ be the value of ζ^{1j} obtained by solving the LP relaxation of ARP-RLT, where the components $(\bar{x}_a, \forall a \in \delta_j^- \cup \delta_j^+)$ within $\bar{\zeta}^{1j}$ are not all binary-valued (since $j \in J_Q$). We would like to generate a valid inequality implied by (3.100)-(3.101) that deletes $\bar{\zeta}^{1j}$, if possible.

Toward this end, denote by π^1 and π^2 the respective dual multipliers associated with Constraints (3.100)-(3.101), where $\pi^2 \geq 0$. Then by duality, or by projecting (3.100)-(3.101) onto the space of the ζ^{1j}-variables (or simply by the process of surrogating (3.100)-(3.101) using the multipliers π^1 and π^2), we have that

$$(\pi^1)^T B\zeta^{1j} \geq 0 \tag{3.102}$$

is a valid inequality, for all $(\pi^1, \pi^2) \in \Pi \equiv \{(\pi^1, \pi^2) : (\pi^1)^T A + (\pi^2)^T D = 0, \pi^2 \geq 0\}$.

Hence, in order to generate a valid inequality of type (3.102) in the space of the original variables ζ^{1j} defining ARP-RLT that deletes $\bar{\zeta}^{1j}$ (if possible), we construct the LP:

$$\text{Minimize} \quad \{(\pi^1)^T B\bar{\zeta}^{1j} : (\pi^1, \pi^2) \in \Pi, (\pi^1)^T B\bar{\zeta}^{1j} \geq -1\}, \tag{3.103}$$

where we have added the constraint $(\pi^1)^T B\bar{\zeta}^{1j} \geq -1$ by way of normalization so that the problem in (3.103) is assured to have an optimum (since it is feasible and bounded from below). The dual to (3.103) is given as follow, where the new variable ζ^0 is the dual variable associated with the normalization constraint in (3.103):

$$\begin{aligned}
\mathbf{LP}^j: \quad & \text{Maximize} && -\zeta^0 && (3.104)\\
& \text{subject to:} && A\zeta^{2j} + (B\bar{\zeta}^{1j})\zeta^0 = B\bar{\zeta}^{1j}, && (3.105)\\
& && D\zeta^{2j} \leq 0, && (3.106)\\
& && \zeta^0 \geq 0. &&
\end{aligned}$$

Let ν^j be the optimal objective function value for LP^j and let (π^{1*}, π^{2*}) denote the optimal dual solution, where π^1 and π^2 are the respective dual variables associated with Constraints (3.105) and (3.106). If $\nu^j = 0$, then there exists a $\bar{\zeta}^{2j}$ such that $(\bar{\zeta}^{1j}, \bar{\zeta}^{2j})$ is feasible to (3.100) and (3.101) and so no inequality of the type (3.102) exists that deletes $\bar{\zeta}^{1j}$. Thus, we skip the consideration of the index $j \in J_Q$. On the other hand, if $\nu^j < 0$, then the valid inequality

$$(\pi^{1*})^T B\zeta^{1j} \geq 0 \tag{3.107}$$

of type (3.102) deletes the current solution $\bar{\zeta}^{1j}$, since by strong duality we have that

$$(\pi^{1*})^T B\bar{\zeta}^{1j} = \nu^j < 0.$$

We therefore add (3.107) to Problem ARP-RLT. Repeating this with each $j \in J_Q$ yields a set of valid inequalities of type (3.107) to further tighten the representation of ARP-RLT. From an empirical perspective, we generated a round of up to 10 cuts of this type using the flights in the selected set J_Q with $Q = 10$. Again, although we can update the LP relaxation and generate another round of such cuts, we did not find this strategy to be computationally advantageous.

In the following section, we present computational results to compare the performance of ARP-RLT, ARP-RLT$^+$, and the model obtained by adding the foregoing set of valid inequalities of type (3.107) to ARP-RLT (we denote this latter lifted model by **ARP-RLT***).

3.4. Computational Results

In this section, we present computational results based on a set of typical test instances that were derived using real flight data obtained from a major US airline company. While commercial mixed-integer program solvers can be directly applied to Model ARP-RLT, we also explore the model augmentation proposed strategies in Sections 3.3.2 and 3.3.3 for further tightening the formulation at the root node based on the initial LP relaxation solution, which yields Models ARP-RLT$^+$ and ARP-RLT*, respectively. Our numerical results demonstrate the efficacy of such strategies. All runs have been made on a 64-bit Windows 7 platform having two Intel Xeon quad-core 2.13GHz CPUs and 4.0 GB memory, where the models have been implemented in ILOG OPL 6.3 and C++ and solved using CPLEX 12.1 with default settings (including the use of the dual simplex method with steepest edge pricing as the LP solver).

3.4.1 Test Cases

The inputs for Problem ARP include a flight schedule with the flight numbers, departure and arrival times, origin and destination nodes, and the flight duration, given a specific fleet type such as Boeing 777, Airbus 320, etc. Moreover, the data sets also specify the aircraft count (number of available aircraft), the minimum turn-time, and the duration of a maintenance check of the predefined aircraft type, as well as a list of maintenance stations that are capable of performing type-A checks. Additionally, we consider the imposed requirements on the total accumulated flying hours, the total number of takeoffs, and the total number of days between consecutive maintenance checks, which are usually taken as stricter limits by airlines than the mandated FAA regulations. It is worth noting that, in practice, airline companies specify a limit on the total number of takeoffs only for overseas flights; hence, for an inland flight schedule (pertaining to North America), constraints related to takeoffs are removed (or a sufficiently large value is assigned to this parameter so as to make the related constraints inactive).

Five practical instances, named *ARP1*, *ARP2*, *ARP3*, *ARP4*, and *ARP5*, were developed to test the performance of the proposed Model ARP-RLT. The first two cases are derived from transcontinental hub-to-hub flights, which are mainly served by wide-body aircraft such as Boeing 777 that require maintenance checks at most every six takeoffs. The flights in Case 3 connect spoke airports to several major hubs by regional jets such as the CRJ series, whose maintenance operations are usually performed only at the hub stations. Each travel duration in this case is comparatively short, and the schedule involves a small sized fleet that provides frequent services between spokes and hubs. Cases 4 and 5 concern flights connecting major cities in the US by mid-sized narrow-body aircraft such as the Airbus 320 series. Since the corresponding flight network covers relatively high demand regions by either direct services or through connections, the related operations require a large-sized fleet as well as several maintenance stations distributed across the network. The salient details of the test instances are presented in Table 3.1. Note that all of these practical test cases involve arcs of types A^1-A^4 only, and so (the typically rare) arc types A^5 and A^6 are omitted from the analysis below.

Table 3.1: Description of the Test Instances

	ARP1	*ARP2*	*ARP3*	*ARP4*	*ARP5*
Number of Flights	28	72	96	166	344
Aircraft Count	21	46	18	52	138
Number of Maintenance Stations	11	11	2	16	16
Minimum Turn-time (*min*)	70	70	34	38	40
A-check Duration (*min*)	360	360	480	420	420
Maximal Flying Time between A-checks (*min*)	3900	3900	4500	2700	2700
Maximal Number of Days between A-checks	4	4	4	4	4
Maximal Number of Take-offs between A-checks	6	6	–	–	–

3.4.2 Results and Analysis

Table 3.2 presents the computational results obtained for the foregoing five test instances, and demonstrates the high efficiency of the proposed formulation ARP-RLT and the other two RLT-enhanced strategies, namely, ARP-RLT$^+$ and ARP-RLT*. Recall that the model classifies all the eligible connections into four different arc types according to whether or not a maintenance opportunity is offered between two consecutive flight legs, and whether the previous flight connects to a same-day or a next-day flight leg. As the number of flights grows, the number of possible connections increases quadratically, which directly affects the number of variables as well as constraints in the model. We also summarize the number of connections of each of the four arc types in Table 3.2. The first four moderately sized instances were solved within 2 CPU seconds; and the largest test case involving a daily schedule with 344 flights was solved in about 10 CPU seconds by all the three strategies based on Model ARP-RLT. The solutions take advantage of almost all the available aircraft in the fleet, as this potentially allows the greatest flexibility in routing. We note here that deriving a tighter LP-based bound for the RHS of Constraint (3.65) did not improve the performance, and so we use the RHS specified in (3.65) in the sequel.

Since the enhanced models ARP-RLT$^+$ and ARP-RLT* are useful only for relatively large-sized data sets, we focus on comparing the performance of these augmented formulations for Instances *ARP3*, *ARP4*, and *ARP5*. Model ARP-RLT$^+$ yielded the best performance with respect to the computational effort for these test instances, except for Instance *ARP4* where Model ARP-RLT* was the most efficient. However, it is relevant to note here that we adopted a simple implementation process for ARP-RLT* whereby, after generating the proposed round of cuts at the root node, we resolve the problem from scratch as opposed to updating the current LP relaxation solution. A more refined implementation would potentially make this strategy relatively more efficient, particularly for larger test instances. For the sake of interest, we also report the number of fractional x-values at the root node in the solution obtained for the LP relaxation corresponding to each of the constructed model representations. It is also worth mentioning that after applying built-in root-node heuristics, an optimal solution was obtained at this initial node itself without further branch-and-cut exploration for all these (feasibility) test cases.

Moreover, we can use Model ARP-RLT (or its augmented versions) to determine the minimum number of aircraft required for a given flight schedule by treating NA as a variable and minimizing it subject to the constraints of Model ARP-RLT. It turns out that, for each of the five instances, the objective value to the underlying LP relaxation yielded the same integer value as that from solving the MIP, even without performing the valid rounding-up operation. The results for this experiment are presented in Table 3.3. For the sake of interest, Table 3.3 also displays the RHS of (3.65) as given by the two preprocessing methods proposed in Proposition 3.3, both of which provide lower bounds on the optimal value of the foregoing MIP. It is evident that γ_1 yields a much tighter lower bound than γ_2 for these instances.

The results from this proposed model cannot be directly compared with those in the existing

Table 3.2: Summary of Results for ARP-RLT, ARP-RLT$^+$, and ARP-RLT*

	ARP1	*ARP2*	*ARP3*	*ARP4*	*ARP5*
# of x_a-variables	158	859	1465	2545	12426
# of A^1 Arcs	7	181	149	445	2484
# of A^2 Arcs	74	495	277	1343	6788
# of A^3 Arcs	71	145	574	569	2453
# of A^4 Arcs	6	38	465	188	701
Results for Model ARP-RLT					
# of Aircraft Utilized	20	46	18	52	134
CPU Time (*sec*)	0.03	0.56	1.62	1.42	9.34
# of Fractional x-values in the LP Relaxation	38	100	123	365	945
Opt. Sol. Obtained at	*root*	*root*	*root*	*root*	*root*
Results for Model ARP-RLT$^+$					
# of Aircraft Utilized	–	–	18	52	136
CPU Time (*sec*)	–	–	0.12	1.26	8.46
# of Fractional x-values in the LP Relaxation	–	–	117	284	964
Opt. Sol. Obtained at	–	–	*root*	*root*	*root*
Results for Model ARP-RLT*					
# of Aircraft Utilized	–	–	18	51	133
CPU Time (*sec*)	–	–	0.17	0.89	10.33
# of Fractional x-values in the LP Relaxation	–	–	125	365	962
Opt. Sol. Obtained at	–	–	*root*	*root*	*root*

Table 3.3: Results for Minimizing the Total Number of Utilized Aircraft

	ARP1	*ARP2*	*ARP3*	*ARP4*	*ARP5*
Obj. Value of MIP	12	41	18	49	87
Obj. Value of LP	12	41	18	49	87
γ_1	12	25	18	36	82
γ_2	11	23	11	28	56

literature since our model explicitly incorporates several operational constraints such as the total flying time, the total number of takeoffs, and the total number of days between two successive maintenance checks, which are not all addressed by any other model. Yet, the results reveal that, even with these additional restrictions, the computational performance for the test instances is still comparable to that reported in the literature. Furthermore, by the structure of Problem ARP, it can be seen that these additional constraints relate the labels on a node to those on its predecessors, similar to the labeling process in the constrained shortest path algorithm that is usually implemented in solving the subproblem of string-based formulations. However, instead of searching for paths with regard to reduced costs/prices, we directly compose feasible paths. Moreover, our compact model permits a direct solution via commercial software, as opposed to string-based models, which require a sophisticated column generation and branch-and-price approach. Specialized implementations and MIP strategies could further enhance the solution efficiency of our model.

It is also worth noting that the model essentially generates cyclic rotations, each of which entails as many aircraft as the number of arcs involved from $A^2 \cup A^4$ plus the number of included wrap-around flights, without actually assigning tail-numbers to such rotations, which is beneficial since it circumvents burdening the model with symmetry-related issues. However, in practice, higher-level maintenance decisions could attempt to balance wear-and-tear on each tail-number by appropriately pulling specific aircraft out of, and inserting them back into, different rotations.

3.4.3 Results on Solving the ARP as an Optimization Problem

In addition, the proposed compact formulation of the ARP could be converted to an optimization-type problem by adopting an objective function that incorporates through-values, short connection penalties, and maintenance costs, the definitions of which were introduced in Chapter 2. A connection is favorable for passengers only if it is convenient, i.e., it does not overnight nor involve long layovers; therefore, through-values are defined subject to connection durations, and are typically assigned only to the arcs in the sets A^3 and (partly) A^4. Note that due to the nature of a through-value, the origin station of the previous flight should not be the same as the destination of the successive flight. Following a similar logic, the short connection penalty is only declared for each connection in the sets A^3 and A^4, which also depends on the time between an arrival and a subsequent departure. These penalties are of interest from the perspective of building robust flight schedules, since it is often desirable to have the slack time between the landing of an aircraft and its next departure long enough so as to absorb any potential delays and therefore mitigate the so-called snowball effect. Based on the connection time, different penalty values can be assigned to connections within $A^3 \cup A^4$; or we can alternatively simply minimize the total number of such critical or short connections (e.g., those having a slack time shorter than 2τ). In addition, maintenance costs are assigned to connections that involve maintenance check opportunities, i.e., connections in the sets A^1, A^2, and A^5. Accordingly, we define $c_a, p_a,$ and m_a as the through-value, the short connection penalty, and the maintenance cost, respectively, for each of the appropriate connections, and formulate the objective function as follows:

$$\text{Maximize} \sum_{a \in A^3 \cup A^4} c_a x_a - \sum_{a \in A^3 \cup A^4} p_a x_a - \sum_{a \in A^1 \cup A^2 \cup A^5} m_a x_a. \tag{3.108}$$

Since total maintenance costs for a given fleet type are roughly constant, we solved our test-bed of Problems *ARP1-ARP5* to optimize (3.108) using the first term with randomly generated coefficients within practical ranges. The performances of Model ARP-RLT based on all the five instances using an optimality tolerance of 0.5% are displayed in Table 3.4, where the actual optimality gap achieved at termination is also presented in the table for each test instance. In addition, we found that, compared with the feasibility model's solution, the solution produced by the optimization model often tends to consolidate all the flights to form a single rotation, which is desirable in that it ensures equal wear-and-tear on all the utilized aircraft. For the sake of interest, we also exhibit in the final row of Table 3.4 the corresponding effort necessary for solving the directly linearized ARP model in the expanded $(x, u, v, d, \omega, \rho, \eta, \theta, \lambda, \mu)$-space (denoted as ARP0-RLT), without the model reduction process prompted by Proposition 3.2. More specifically, this expanded space formulation is given as follows:

$$\begin{array}{lll} \textbf{ARP0-RLT:} & \text{Find} & x, u, v, d, \omega, \rho, \eta, \theta, \lambda, \mu \\ & \text{subject to:} & \\ & & (3.2), (3.3), (3.12), (3.16), (3.20)\text{-}(3.25), \text{ and } (3.32)\text{-}(3.46). \end{array} \tag{3.109}$$

The results in Table 3.4 demonstrate a significant improvement in performance for Models ARP-RLT, ARP-RLT$^+$, and ARP-RLT* over Model ARP0-RLT, which reveals, in particular, the relative effectiveness of the LP-equivalent formulation ARP-RLT derived via Proposition 3.2 over the higher dimensional representation ARP0-RLT. Moreover, the two RLT-enhanced formulations also outperformed Model ARP-RLT for the large data sets. Furthermore, for the sake of interest, we also report the number of fractional x-values in the LP relaxation solutions for each of the formulated models ARP-RLT, ARP-RLT$^+$, and ARP-RLT*. Moreover, we provide in parentheses the residual number of unresolved fractional variables at the root node after CPLEX has performed its built-in additional preprocessing steps including the derivation of heuristic solutions.

Overall, Models ARP-RLT$^+$ and ARP-RLT*, enhanced by the partial convexification process, outperformed Model ARP-RLT in solving both feasibility and optimization problems. Indeed, the partial convex hull representations constructed by Model ARP-RLT$^+$ as well as ARP-RLT* tighten the underlying LP relaxation, facilitate the root node processing, and hence promote a more efficient solution process for relatively challenging problems. In general, Model ARP-RLT* has fewer constraints than ARP-RLT$^+$; but in our test runs, Model ARP-RLT$^+$ always solved the problems quite effectively at the root node itself. However, as indicated above, a more refined implementation of ARP-RLT* might make it relatively more advantageous for larger test cases. This is open to further investigation in future research.

Table 3.4: Computational Results Using the Objective Function (3.108)

	ARP1	*ARP2*	*ARP3*	*ARP4*	*ARP5*
Results for Model ARP0-RLT					
# of Aircraft Utilized	14	44	18	52	107
Opt. Gap	0.28%	*opt*	*opt*	0.38%	0.12%
CPU Time (*sec*)	1.03	1.03	1.00	6.29	10.67
# of Fractional x-values					
in the LP Relaxation	14(0)	42(0)	18(0)	60(18)	199(0)
# of Nodes Enumerated	*root*	*root*	*root*	57	*root*
Results for Model ARP-RLT					
# of Aircraft Utilized	14	44	18	52	105
Opt. Gap	*opt*	*opt*	*opt*	0.42%	0.08%
CPU Time (*sec*)	0.25	0.55	0.86	4.60	8.91
# of Fractional x-values					
in the LP Relaxation	10(0)	34(0)	29(0)	78(0)	136(0)
# of Nodes Enumerated	*root*	*root*	*root*	*root*	*root*
Results for Model ARP-RLT$^+$					
# of Aircraft Utilized	–	–	18	52	106
Opt. Gap	–	–	*opt*	0.35%	*opt*
CPU Time (*sec*)	–	–	0.11	2.12	3.46
# of Fractional x-values					
in the LP Relaxation	–	–	22(0)	76(0)	169(0)
# of Nodes Enumerated	–	–	*root*	*root*	*root*
Results for Model ARP-RLT*					
# of Aircraft Utilized	–	–	18	52	109
Opt. Gap	–	–	*opt*	0.49%	0.11%
CPU Time (*sec*)	–	–	0.09	3.45	6.90
# of Fractional x-values					
in the LP Relaxation	–	–	18(0)	83(39)	170(0)
# of Nodes Enumerated	–	–	*root*	426	*root*

3.5. Summary and Conclusions

In this chapter, we have presented a new formulation for the daily aircraft maintenance routing problem. Distinct from path-based formulations that comprise exponentially many variables, we have designed a novel node-arc flow formulation, which is of polynomial size, and thus yields a more compact representation. The model inherits certain direct, simple nonlinear relationships, which were reformulated, linearized, and lifted using the RLT approach to produce an improved linear zero-one mixed-integer programming formulation that can be directly solved using available commercial software. This model can be used by airline companies to generate feasible routes for a daily repetitive schedule of a given fleet, as well as optimized rotations based on specified through-values, short connection penalties, and maintenance costs.

In parallel to solving Model ARP-RLT directly, we have also developed two root-node strategies for further enhancing the proposed model representation based on a partial convexification process. Computational results reveal that the resulting enhanced formulations ARP-RLT$^+$ and ARP-RLT*, which either explicitly construct certain partial convex hull representations or derive suitable valid inequalities implied by this augmented structure, respectively, can be solved efficiently both as feasibility and optimization problems using default settings of a commercial MIP solver, and provide a computational advantage over the basic lifted model.

The proposed ARP model can be readily modified to accommodate several specific features of interest to the airline industry. For example, an alternative requirement on aircraft routing is to make decisions for a specific period (e.g., a week) given the start station and the u-, v-, and d-labels for each aircraft. To account for this requirement, we could generate paths (instead of rotations) through the prescribed period while keeping track of these labels in a similar fashion. Moreover, to better fit industrial practice, we could also incorporate other types of relatively frequent checks as necessary within the model by introducing multiple-service or parallel single-service connection arcs with respective counters to account for the corresponding durations between two consecutive maintenances. In addition, to assess the robustness of the generated solutions, we can examine the realized values of the u-, v-, and d-labels relative to their respective specified maximal values. Note that the solutions could be made more robust by providing tighter maximal permissible values of these parameters.

We can also include additional aggregate constraints to ensure that at least a certain percentage of aircraft undergo maintenance operations on a daily basis by requiring the sum of maintenance arcs included over all generated cycles to be at least a certain percentage of NA. Finally, to model possible retiming of flights, we can create copies of flight nodes corresponding to the alternative retimed schedules along with appropriate flight connections, and include a set of constraints that requires the coverage of exactly one copy of each flight (see Sherali et al. (2011) for modeling this feature within the context of fleet assignment problems).

We now proceed to embed the developed ARP model within a more extensive integrated model of the type discussed in Section 2.1, which further considers the related airline operational problems of fleet assignment and crew pairing.

Chapter 4

An Integrated Model for Aircraft Fleeting, Routing, and Crew Pairing

4.1. Introduction

The airline scheduling problem deals with maximizing the total profits or minimizing the total operational costs associated with a flight schedule while satisfying customer demands under a series of restrictions concerning aircraft maintenance and labor work-rules. Specifically, the airline scheduling problem can be decomposed into four stages: *schedule planning* (designing flight schedules), *fleet assignment* (assigning aircraft fleets to flight legs), *aircraft routing* (generating maintenance-feasible rotations for aircraft of each type), and *crew pairing* (generating suitable flight sequences for crews to serve), where the output of one stage is fed into the succeeding stage in a sequential implementation.

After the flight schedule is determined, fleets of different types of aircraft are assigned to different flight legs to maximize the fare-based revenue minus the burned fuel cost and the cost for *spilled* (or lost) demand depending on the assigned fleet's capacity. Here, an *aircraft type* refers to a certain model of aircraft. All aircraft of the same type are identical; that is, they have the same cockpit configuration, crew rating, and capacity. Furthermore, an *aircraft family* is a set of aircraft types having the same cockpit configuration and crew rating, but different seating capacities. The outcome of the fleet assignment stage partitions the flight network into sub-networks corresponding to aircraft types or families, so that the following two stages of aircraft routing and crew pairing are solved for each specific type or family of aircraft, respectively.

The aircraft routing problem, which is solved next, finds cyclic *maintenance feasible routes* for the aircraft in the fleet while satisfying mandatory maintenance checks, with the objective of optimizing a selective combination of *through-values* (additional revenue accruing from the same aircraft serving consecutive flight legs), *short connection penalties* (additional cost due to insufficient intermediate rest or *sit-time* for crews), and/or maintenance costs. We note that, in many cases, the aircraft routing problem is treated as a pure feasibility problem. In addition, a maintenance feasible aircraft route (or *rotation*) is comprised of a sequence of flight legs that are served by a single aircraft and that satisfy the following restrictions:

- the departure station of each flight leg must match with the arrival station of the preceding flight leg;
- the rotation must include at least one visit to a maintenance station, where the time elapsed between the arrival time of the flight leg to this station and the subsequent departure time of the next flight must exceed the time required to perform the mandated maintenance check;
- the total flying time between maintenance checks should not exceed a specified limit;
- the total number of takeoffs between maintenance checks should not exceed a specified maximum value;
- the number of days elapsed between maintenance checks must not exceed a given number of calendar days.

The crew pairing stage requires building a minimum-cost set of pairings such that each leg is covered at least once (note that a leg may be multi-covered since pairings may include *deadheads*, i.e., flight legs for relocating crews). Since cockpit crews are qualified to serve only one aircraft family, the crew pairing problem is solved separately for each aircraft family. In this context, a *duty period* refers to a single workday of a crew that is comprised of serving a sequence of flight legs with short rest periods, or *sits*, separating them, including briefing and debriefing periods at the beginning and end of the period. Crews are assigned to a sequence of flight legs subject to several work-rule requirements established by the *Federal Aviation Administration* (FAA) and union contracts (see Klabjan (2005)). For instance, a non-exhaustive list of rules includes the following restrictions, where typical numerical values are specified for the sake of illustration:

- There should be no more than four flight legs in a duty;
- The *minimum sit-time* should be at least 45 minutes. However, if both flight legs are covered by the *same* aircraft, then the sit-time could be reduced to the predefined minimum short connection duration, which is typically 30 minutes (note that this requirement relates aircraft routing decisions to crew scheduling decisions);
- The maximum sit-time should not exceed four hours;
- The total flying time within a duty period must not exceed eight hours;
- The total duration of a duty period must not exceed 12, 13, or 14 hours, depending on the duty start-time;
- The duration of a rest period (or *layover*) following a duty period should be greater than or equal to the maximum of: 10 hours, duty period duration, and twice the total flying time within the particular duty period;
- The number of duty periods in a pairing must be at least one and at most four;
- The total flying time in a pairing should not exceed 30 hours;
- The sum of the durations of the duty periods that constitute a pairing should not exceed 56 hours.
- The total *time away from base* (TAFB) should not exceed 96 hours.

The cost of a duty period, expressed in hours, is the maximum of three quantities: a certain fraction of the duty period duration, the total flying time, and a specified minimum number of hours. Similarly, the cost of a pairing, expressed in hours, is the maximum of three quantities: a certain fraction of the pairing duration, the sum of the costs of the duty periods comprising the pairing, and the specified minimum number of hours per duty period times the number of duty periods in the pairing.

Although the foregoing operational phases are *implemented* sequentially in practice, their interdependence would naturally lead a purely sequential *decision-making* approach to sub-optimal solutions, because prefixed decisions that are made while ignoring downstream considerations would tend to sub-optimally restrict the ensuing decision stages. This might even result in infeasibility at some subsequent stages in the process. From this perspective, it is prudent to investigate models that integrate the different stages (or suitable combinations thereof) within a single framework in order to obtain improved solutions, while being cognizant of the fact that the problem complexity will also substantially increase as more aspects and decisions are considered simultaneously.

In this chapter, we propose novel modeling and algorithmic approaches for solving the integrated airline scheduling model that simultaneously considers the operational stages of fleet assignment, aircraft routing, and crew pairing. This integration is crucial for airlines not only because of the high fleeting and crew costs, but also because these three processes are closely interacting components of the planning process. Therefore, we formulate an integrated model to maximize the total profit by assigning each aircraft (or *tail number*) to each flight leg, and to determine compatible sets of maintenance-feasible aircraft routings, along with work-rule feasible sets of crew pairings to serve corresponding sequences of flight legs, where we are given as inputs the flight schedule, the regulations on aircraft maintenance requirements, and the agreements on crew work-rules. The proposed model also accommodates a more realistic specification of itinerary-based demands, and is tested using real data from a major US carrier (*United Airlines*). For effectively solving this integrated problem, the model is decomposed using a Benders approach into a master program involving aircraft fleeting and routing decisions, and two subproblems that respectively address itinerary-based passenger-mix and crew pairing decisions. Moreover, for solving the crew pairing subproblem, we adopt a specialized perturbed Lagrangian dual approach, as recommended by Subramanian and Sherali (2008), which is designed to avoid a frequently observed stalling phenomenon, and we embed this within a branch-and-price heuristic. In addition, several acceleration techniques are implemented along with a specialized deflected subgradient optimization scheme to better tackle the master program as well as the subproblems.

Specifically, this chapter makes the following contributions:

1. We propose a novel formulation for the integrated airline scheduling model that incorporates aircraft fleeting, routing, and crew pairing operations within a single framework, in which the fleeting and routing decisions are modeled using a polynomially-sized node-arc flow network representation, and the crew pairing decisions are modeled using the traditional set partitioning approach. In addition, we incorporate within the model several realistic operational considerations, such as itinerary-based demands and various mandated aircraft maintenance requirements and crew work-rules.

2. We design an effective solution strategy for the developed large-scale model using a Benders decomposition approach that incorporates nondominated Benders cuts along with several acceleration techniques in order to enhance solvability. Moreover, for the crew pairing subproblem, we adopt a perturbed Lagrangian dual approach along with a specialized deflected subgradient optimization scheme for stabilizing the column generation process, and we embed this within a branch-and-price framework.

3. We provide extensive computational results using realistic data obtained from a large-sized US-based airline company in order to demonstrate the efficacy of our modeling approach and solution methodology, and we exhibit the benefits of adopting an integrated approach as opposed to a sequential decision process. The results reveal that this yields an average of 2.73% improvement in profits, which roughly translates to 43 million dollars per year.

The remainder of this chapter is organized as follows. In Section 4.2, we present our proposed integrated formulation. Following this, we develop several model enhancement techniques in Section 4.3, and design a suitable solution strategy. Computational results using realistic simulated data are presented and discussed in Section 4.4, and we close the chapter in Section 4.5 with some concluding remarks and future research directions.

4.2. Model Formulation

Prior to presenting our integrated model, we introduce the following notation based on a *flight connection network* involving a set of daily flights:

Notation

Let L denote the set of daily flight legs under consideration. For each flight leg $j \in L$, we define the following notation (with all time durations expressed in minutes):

$DT_j \in [0, 1440]$: departure time of flight leg j.

$AT_j \in [0, 1440]$: arrival time of flight leg j.

DS_j: departure station of flight leg j.

AS_j: arrival station of flight leg j.

t_j: flying time of flight leg j (we assume that $t_j \leq 1440$ for practical reasons).

na: number of *wrap-around flights* in the schedule (i.e., flights that depart on a given day, cross the end of the time horizon, and arrive on the next day).

$L_w \equiv \{j \in L : j \text{ is a wrap-around flight}\}$.

Let K denote the set of all aircraft types. For each aircraft of type $k \in K$, we define:

τ^k: turn-time for aircraft of type $k, \forall k \in K$.

NA^k: number of available aircraft of type $k, \forall k \in K$.

$t^k_{\max}$: maximum flying time between two consecutive maintenance checks for aircraft of type k (note that $t_j \leq t^k_{\max}, \forall j \in L, k \in K$).

$to^k_{\max}$: maximum number of takeoffs between two consecutive maintenance checks for aircraft of type k ($to^k_{\max} \geq 1, \forall k \in K$).

$d^k_{\max}$: maximum number of days between two consecutive maintenance checks for aircraft of type k ($d^k_{\max} \geq 1, \forall k \in K$).

M^k: time duration of a maintenance check for aircraft of type $k, \forall k \in K$ ($\tau^k < M^k < 1440$).

S^k: set of maintenance stations for aircraft of type $k, \forall k \in K$ ($|S^k| \geq 1$).

For a given flight schedule, we define an associated digraph $G = (V, A)$ in which each node $j \in V$ represents a flight leg. Furthermore, for each aircraft of type $k \in K$, we define a corresponding arc set A^k, where $A \equiv \cup_{k \in K} A^k$, and where each arc $a \in A^k$ represents a feasible connection, i.e., an arc $a \in A^k$ if and only if an aircraft of type k can consecutively serve the flights pertaining to the from-node and the to-node of this arc, respectively denoted as a^- and a^+. Also, for notational convenience, we denote the set of arcs that are incident to, and that are outgoing from, node $j \in V$ by δ_j^- and δ_j^+, respectively. More specifically, following the development in Haouari et al. (2011a), the set of arcs A^k, for each $k \in K$, is given by the union of six arc subsets A_1^k, A_2^k, A_3^k, A_4^k, A_5^k, and A_6^k that are defined as follows, where for the sake of ease in reading, we omit the superscript k for related sets and parameters below:

- An arc $a \in A_1$ if and only if a maintenance check could be planned between the arrival of flight leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and both flight legs are required to be served consecutively on the same day. Hence, $(j, l) \in A_1 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \in S$; and (iii) $AT_j + M \leq DT_l$.

- An arc $a \in A_2$ if and only if a maintenance check could be planned between the arrival of flight leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and the same aircraft is required to serve flight leg l the day after serving flight leg j, even if $AT_j + \tau \leq DT_l$. Hence, $(j, l) \in A_2 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \in S$; and (iii) $DT_l < AT_j + M \leq DT_l + 1440$.

- An arc $a \in A_3$ if and only if a maintenance check could not be planned between the arrival of flight leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and both flight legs are required to be served consecutively on the same day. Hence, $(j, l) \in A_3 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \notin S$ or $DT_l < AT_j + M$; and (iii) $AT_j + \tau \leq DT_l$.

- An arc $a \in A_4$ if and only if a maintenance check could not be planned between the arrival of flight leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and the same aircraft serves flight leg l the day after serving flight leg j. Hence, $(j, l) \in A_4 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \notin S$ or $AT_j + M > DT_l + 1440$; and (iii) $DT_l < AT_j + \tau \leq DT_l + 1440$.

- An arc $a \in A_5$ if and only if a maintenance check could be planned between the arrival of leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and the same aircraft is required to serve flight leg l two days after serving flight leg j. This type of arc represents the situation in which the aircraft does not have enough time to undergo maintenance after serving flight j and then to serve flight l the next day. Hence, $(j, l) \in A_5 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \in S$; and (iii) $DT_l + 1440 < AT_j + M \leq DT_l + 2880$.

- An arc $a \in A_6$ if and only if a maintenance check could not be planned between the arrival of flight leg $a^- \equiv j$ and the departure of flight leg $a^+ \equiv l$, and the same aircraft is required to serve flight leg l two days after serving flight leg j. Hence, $(j, l) \in A_6 \Leftrightarrow$ (i) $AS_j \equiv DS_l$; (ii) $AS_j \notin S$; and (iii) $DT_l + 1440 < AT_j + \tau \leq DT_l + 2880$.

A *cycle* $\{j_1, j_2, \ldots, j_p, j_1\}$ in G corresponds to an aircraft rotation that consecutively covers flight legs $j_1, j_2, \ldots, j_p$ and back to j_1 in a cyclic fashion. If a cycle includes ξ_1 arcs belonging to $A_2 \cup A_4$ and ξ_2 arcs belonging to $A_5 \cup A_6$, and also covers ξ_3 wrap-around flights, then the corresponding rotation spans $\xi = \xi_1 + 2\xi_2 + \xi_3$ consecutive days, and since each flight leg is scheduled every day, ξ aircraft should be assigned to this rotation with each aircraft serving this same sequence over ξ days, but staggered so that the union of the flights served by each of these ξ aircraft each day equals $(j_1, \ldots, j_p)$ while their intersection is empty. Also, if a rotation includes an arc $a \in A_1 \cup A_2 \cup A_5$, then an aircraft assigned to this rotation has the opportunity to (but may not necessarily) undergo a maintenance check between flight legs a^- and a^+.

Furthermore, in order to incorporate itinerary-based flight demands, we define the following notation:

Π: set of all itineraries.

$\Pi_j \subset \Pi$: the subset of itineraries that include flight $j, \forall j \in L$.

H: set of all fare-classes.

γ_{kh}: capacity of aircraft of type $k \in K$ to accommodate passengers for fare-class $h \in H$.

μ_{ph}: mean demand for fare-class $h \in H$ within itinerary $p \in \Pi$.

f_{ph}: estimated revenue for fare-class $h \in H$ within itinerary $p \in \Pi$.

r_a: estimated through-value for connection $a \in A$.

In addition, we adopt the same fleet assignment cost definition as used in Mansour et al. (2010), which incorporates the fixed operating charges as well as the opportunity cost due to *spilled passengers*. Accordingly, we denote the cost associated with assigning an aircraft of type k to flight leg j as c_{jk}, which can be expressed as

$$c_{jk} = \bar{c}_{jk} + \sum_{h \in H} o_{jh} \big(\sum_{p \in \Pi_j} \mu_{ph} - \gamma_{kh} \big)^+, \tag{4.1}$$

where $\bar{c}_{jk}$ denotes the fixed cost of assigning fleet type k to flight j, and o_{jh} represents the opportunity cost per *spilled passenger* on flight j that is incurred where the expected demand for fare-class h exceeds the capacity of the assigned aircraft, and where $(\cdot)^+ \equiv \max\{0, \ \cdot\}$.

Decision Variables:

x_a: binary variable that equals 1 if arc $a \in A$ is selected, and 0 otherwise.

y_{jk}: binary variable that equals 1 if flight leg $j \in L$ is assigned to an aircraft of type k, and 0 otherwise.

u_j^k: total accumulated flying hours for aircraft of type $k \in K$ since its last maintenance check after serving flight leg $j \in L$.

v_j^k: total number of takeoffs for aircraft of type $k \in K$ since its last maintenance check after serving flight leg $j \in L$.

d_j^k: total number of elapsed days for aircraft of type $k \in K$ since the last maintenance check after serving flight leg $j \in L$.

π_{ph}: number of passengers accepted for itinerary $p \in \Pi$ within fare-class $h \in H$.

Based on the above notation, we first formulate an integrated *fleet assignment and aircraft routing* (**FAAR**) problem as follows, where for convenience, we denote the set of maintenance-permitting arcs as $A_M^k \equiv A_1^k \cup A_2^k \cup A_5^k$, and the set of non-maintenance arcs as $A_{NM}^k \equiv A_3^k \cup A_4^k \cup A_6^k$, for each $k \in K$:

$$\textbf{FAAR:} \quad \text{Maximize} \sum_{p \in \Pi} \sum_{h \in H} f_{ph} \pi_{ph} + \sum_{a \in A} r_a x_a - \sum_{j \in L} \sum_{k \in K} c_{jk} y_{jk} \tag{4.2}$$

subject to:

$$\sum_{k\in K} y_{jk} = 1, \qquad \forall j \in L, \tag{4.3}$$

$$\sum_{a\in\delta_j^- \cap A^k} x_a = y_{jk}, \qquad \forall j \in L, k \in K, \tag{4.4}$$

$$\sum_{a\in\delta_j^+ \cap A^k} x_a = y_{jk}, \qquad \forall j \in L, k \in K, \tag{4.5}$$

$$u_j^k x_a = t_j x_a, \qquad \forall a \in \delta_j^- \cap A_M^k, j \in L, k \in K, \tag{4.6}$$

$$u_j^k x_a = (u_{a^-}^k + t_j) x_a, \qquad \forall a \in \delta_j^- \cap A_{NM}^k, j \in L, k \in K, \tag{4.7}$$

$$v_j^k x_a = x_a, \qquad \forall a \in \delta_j^- \cap A_M^k, j \in L, k \in K, \tag{4.8}$$

$$v_j^k x_a = (v_{a^-}^k + 1) x_a, \qquad \forall a \in \delta_j^- \cap A_{NM}^k, j \in L, k \in K, \tag{4.9}$$

$$d_j^k x_a = x_a, \qquad \forall a \in \delta_j^- \cap A_M^k, j \in L, k \in K, \tag{4.10}$$

$$d_j^k x_a = d_{a^-}^k x_a, \qquad \forall a \in \delta_j^- \cap A_3^k, j \in L, k \in K, \tag{4.11}$$

$$d_j^k x_a = (d_{a^-}^k + 1) x_a, \qquad \forall a \in \delta_j^- \cap A_4^k, j \in L, k \in K, \tag{4.12}$$

$$d_j^k x_a = (d_{a^-}^k + 2) x_a, \qquad \forall a \in \delta_j^- \cap A_6^k, j \in L, k \in K, \tag{4.13}$$

$$\sum_{a\in A_2^k \cup A_4^k} x_a + 2\sum_{a\in A_5^k \cup A_6^k} x_a + \sum_{j\in L_w} y_{jk} \le NA^k, \qquad \forall k \in K, \tag{4.14}$$

$$t_j \le u_j^k \le t_{\max}^k, \qquad \forall j \in L, k \in K, \tag{4.15}$$

$$1 \le v_j^k \le to_{\max}^k, \qquad \forall j \in L, k \in K, \tag{4.16}$$

$$1 \le d_j^k \le d_{\max}^k, \qquad \forall j \in L, k \in K, \tag{4.17}$$

$$\sum_{p\in\Pi_j} \pi_{ph} \le \sum_{k\in K} \gamma_{kh} y_{jk}, \qquad \forall j \in L, h \in H, \tag{4.18}$$

$$0 \le \pi_{ph} \le \mu_{ph}, \text{ integer}, \qquad \forall p \in \Pi, h \in H, \tag{4.19}$$

$$(x,\ y) \text{ binary}. \tag{4.20}$$

The objective function (4.2) maximizes the overall profit given by the total revenue, including through-values, minus the fleet assignment costs. Constraint (4.3) requires each flight leg to be covered by exactly one fleet type. Constraints (4.4) and (4.5) ensure that each flight has exactly one predecessor and one successor, respectively, both of which are assigned to the same aircraft type. Hence, together with x being binary-valued, these restrictions induce the solution to be comprised of cycles or cyclic rotations, where each rotation covers a set of flights using a particular type of aircraft. Note that since each rotation describes a cyclic use of a particular aircraft of some type, the traditional conservation of flow constraints at stations that are written for typical fleet assignment models using a time-space network representation are automatically satisfied (see Sherali et al. (2006), for example). For each aircraft type, the nonlinear constraints (4.6)-(4.7) together with (4.15) enforce the total flying time restrictions. Note that the nature of these cyclic constraints precludes a rotation having no maintenance visit. Similarly, Constraints (4.8)-(4.9) and (4.16) assure the restrictions on the maximal number of takeoffs, and Constraints (4.10)-(4.13) and (4.17) likewise guarantee the restrictions on the maximum number of days between maintenance checks. Observe that, given binary values of (x, y) feasible to Constraints (4.3)-(4.5), we have that Constraints (4.8)-(4.13) automatically induce the v- and d-variables to be integer-valued; hence, these variables are logically declared in Constraints (4.16) and (4.17) to be simply continuous-valued. (In this same vein, if t_j are integer-valued, then so are the u-variable values in any feasible solution.) Constraint (4.14) requires that the total number of aircraft in service, as accounted for within each cyclic rotation as discussed above, should not exceed the available size of the fleet (where the sum of the ξ_3-values over all $k \in K$ equals $\sum_{k \in K} \sum_{j \in L_w} y_{jk} = |L_w| = na$ by (4.3)). Note that this is similar to an aircraft count constraint that is used in fleet assignment models (e.g., see Sherali et al. (2006)), where the accounting in this case is done at the count time-line $t = 1440$, and where the first two terms in the left-hand side (LHS) of (4.14) count the number of aircraft of type k on the ground, and where the third term equals the number of aircraft of type k in the air at that time. Constraint (4.18) enforces that the total number of passengers traveling on a flight within a specific fare-class is no more than the capacity available for that fare-class on the assigned aircraft. Constraint (4.19) requires that the number of passengers accepted on any particular itinerary for each fare-class does not exceed the corresponding expected demand. Finally, Constraint (4.20) imposes logical restrictions on the binary decision variables.

Following Sherali et al. (2010), we can replace (4.18) by the tighter inequality:

$$\sum_{p \in \Pi_j} \pi_{ph} \leq \sum_{k \in K} \tilde{\gamma}_{jkh} y_{jk}, \qquad \forall j \in L, h \in H, \tag{4.21}$$

where $\tilde{\gamma}_{jkh} \equiv \min\{\gamma_{kh}, \sum_{p \in \Pi_j} \mu_{ph}\}, \forall j \in L, k \in K, h \in H$.

Moreover, we can replace (4.19) by

$$0 \leq \pi_{ph} \leq \tilde{\mu}_{ph} \equiv \min\{\mu_{ph}, \max_{k\in K}\gamma_{kh}\}, \forall p \in \Pi, h \in H. \tag{4.22}$$

Remark 4.1: The y-variables can be eliminated from the formulation by substituting them in terms of the x-variables using Constraint (4.4) or (4.5). Practically, in order to induce sparsity in the resulting model, we can substitute y_{jk} with either of the expressions on the LHS of (4.4) and (4.5), whichever yields an overall sparser constraint (4.3), for each $j \in L$. Arbitrarily selecting (4.4) in this context, Constraints (4.3)-(4.5), (4.14), and (4.21) can be replaced by the following:

$$\sum_{a\in\delta_j^-} x_a = 1, \qquad \forall j \in L, \tag{4.23}$$

$$\sum_{a\in\delta_j^-\cap A^k} x_a = \sum_{a\in\delta_j^+\cap A^k} x_a, \qquad \forall j \in L, k \in K, \tag{4.24}$$

$$\sum_{a\in A_2^k\cup A_4^k} x_a + 2\sum_{a\in A_5^k\cup A_6^k} x_a + \sum_{j\in L_w}\sum_{a\in\delta_j^-\cap A^k} x_a \leq NA^k, \qquad \forall k \in K, \tag{4.25}$$

$$\sum_{p\in\Pi_j} \pi_{ph} \leq \sum_{k\in K}\sum_{a\in\delta_j^-\cap A^k} \tilde{\gamma}_{jkh}x_a, \qquad \forall j \in L, h \in H, \tag{4.26}$$

and the objective function can be rewritten as follows:

$$\text{Maximize} \qquad \sum_{p\in\Pi}\sum_{h\in H} f_{ph}\pi_{ph} - \sum_{j\in L}\sum_{k\in K}\sum_{a\in\delta_j^-\cap A^k} (c_{jk} - r_a)x_a. \ \Box \tag{4.27}$$

4.2.1 Reformulation and Linearization

To enhance the solvability of the integrated model, we next propose to apply the special-structured Reformulation-Linearization Technique (RLT) of Sherali et al. (1998) to derive a tight, equivalent linear model representation of this problem. Toward this end, first consider Constraints (4.6) and (4.7). Following Haouari et al. (2011a), we can linearize these constraints by using the substitutions:

$$\omega_{jl}^k = u_j^k x_{jl}, \qquad \forall (j,l) \in A^k, k \in K, \tag{4.28}$$

$$\rho_{jl}^k = u_l^k x_{jl}, \qquad \forall (j,l) \in A^k, k \in K. \tag{4.29}$$

Accordingly, Constraints (4.6) and (4.7) become

$$\rho^k_{jl} = t_l x_{jl}, \qquad \forall (j,l) \in A^k_M, k \in K, \tag{4.30}$$

$$\rho^k_{jl} = \omega^k_{jl} + t_l x_{jl}, \qquad \forall (j,l) \in A^k_{NM}, k \in K. \tag{4.31}$$

Now, in order to assure the product relationships (4.28) and (4.29) based on the RLT process, consider the following set that is implied by (4.23), (4.24), and (4.20):

$$SS^k_j = \{x : \sum_{a \in \delta^-_j \cap A^k} x_a = \sum_{a \in \delta^+_j \cap A^k} x_a, \sum_{a \in \delta^-_j \cap A^k} x_a \leq 1, x_a \geq 0, \forall a \in (\delta^-_j \cup \delta^+_j) \cap A^k\}, \forall j \in L, k \in K.$$

Since SS^k_j implies that $\{0 \leq x_a \leq 1, \forall a \in (\delta^-_j \cup \delta^+_j) \cap A^k\}, \forall j \in L, k \in K$, it suffices to take inter-products of the constraints defining SS^k_j with (4.15) for each $j \in L, k \in K$ for the purpose of enforcing (4.28) and (4.29) while deriving a tight linear programming (LP) representation. Hence, for each $j \in L, k \in K$, multiplying Constraint (4.15) by $x_{lj}, \forall (l,j) \in A^k$, and by $x_{jl}, \forall (j,l) \in A^k$, we respectively obtain the following inequalities upon linearization (where we have interchanged the indices l and j for the first case):

$$t_l x_{jl} \leq \rho^k_{jl} \leq t^k_{\max} x_{jl}, \qquad \forall (j,l) \in A^k, k \in K, \tag{4.32}$$

$$t_j x_{jl} \leq \omega^k_{jl} \leq t^k_{\max} x_{jl}, \qquad \forall (j,l) \in A^k, k \in K. \tag{4.33}$$

Next, multiplying the first constraint in SS^k_j by simply u^k_j (noting that this is an equality restriction), we derive the following constraint upon linearization using (4.28) and (4.29):

$$\sum_{l:(l,j) \in A^k} \rho^k_{lj} = \sum_{l:(j,l) \in A^k} \omega^k_{jl}, \forall j \in L, k \in K. \tag{4.34}$$

Finally, multiplying the second constraint in SS^k_j with the two bound-factors in (4.15) lifts the latter (noting (4.32)) to the following restrictions upon linearization using (4.28) and (4.29):

$$t_j + \sum_{l:(l,j) \in A^k} (\rho^k_{lj} - t_j x_{lj}) \leq u^k_j \leq t^k_{\max} - \sum_{l:(l,j) \in A^k} (t^k_{\max} x_{lj} - \rho^k_{lj}), \forall j \in L, k \in K. \tag{4.35}$$

Thus, by Sherali et al. (1998), we can derive an equivalent lifted model by replacing (4.6), (4.7), and (4.15) in Problem with (4.30)-(4.35). However, observe that the variable u^k_j

appears only in (4.35) within the resulting model, and so this constraint can be eliminated and the u-variables can be determined *posteriori* using (4.35) so long as

$$t_j + \sum_{l:(l,j)\in A^k} (\rho^k_{lj} - t_j x_{lj}) \leq t^k_{\max} - \sum_{l:(l,j)\in A^k} (t^k_{\max} x_{lj} - \rho^k_{lj}), \forall j \in L, k \in K, \text{ i.e.,}$$

$$t_j(1 - \sum_{l:(l,j)\in A^k} x_{lj}) \leq t^k_{\max}(1 - \sum_{l:(l,j)\in A^k} x_{lj}), \forall j \in L, k \in K,$$

which automatically holds since $t_j \leq t^k_{\max}$ and $\sum_{l:(l,j)\in A^k} x_{lj} \leq 1$ by (4.23). Thus, we can drop (4.35) and replace (4.6), (4.7), and (4.15) with (4.30)-(4.34).

By using (4.30) and (4.31) to further eliminate the ρ-variables from (4.30)-(4.34), we can readily establish the following proposition (which is similar to Proposition 2 in Haouari et al. (2011a)):

Proposition 4.1. *Constraints (4.30)-(4.34) can be replaced by the restrictions (4.36)-(4.37) given below, which, together with (4.23) and (4.24), yield an equivalent set of constraints, even in the sense of the continuous LP relaxation:*

$$\sum_{l:(j,l)\in A^k} \omega^k_{jl} = t_j \sum_{l:(l,j)\in A^k} x_{lj} + \sum_{l:(l,j)\in A^k_{NM}} \omega^k_{lj}, \qquad \forall j \in L, k \in K, \tag{4.36}$$

$$t_j x_{jl} \leq \omega^k_{jl} \leq b^{t^k}_{jl} x_{jl}, \qquad \forall (j,l) \in A^k, k \in K, \tag{4.37}$$

where $b^{t^k}_{jl} = \begin{cases} t^k_{\max} - t_l, \ \forall (j,l) \in A^k_{NM}, \\ t^k_{\max}, \ \forall (j,l) \in A^k_M. \end{cases}$ □

The nonlinear constraints (4.8)-(4.13) can be lifted and linearized following an identical derivation by defining the new set of RLT variables:

$$\eta^k_{jl} = v^k_j x_{jl}, \qquad \forall (j,l) \in A^k, k \in K, \tag{4.38}$$

$$\lambda^k_{jl} = d^k_j x_{jl}, \qquad \forall (j,l) \in A^k, k \in K, \tag{4.39}$$

and by introducing the following parameters for each $k \in K$, similar to the parameters $b^{t^k}_a, a \in A$, defined in Proposition 1 (written using the notation $a \in A^k$ in lieu of $(j,l) \in A^k$):

$$b^{to^k}_a = \begin{cases} to^k_{\max} - 1, \ \forall a \in A^k_{NM}, \\ to^k_{\max}, \ \forall a \in A^k_M, \end{cases}$$

$$b_a^{d^k} = \begin{cases} d_{\max}^k - 1, \ \forall a \in A_4^k, \\ d_{\max}^k - 2, \ \forall a \in A_6^k, \\ d_{\max}^k, \ \forall a \in A_M^k \cup A_3^k. \end{cases}$$

This yields the following equivalent reformulation of Problem FAAR:

$$\textbf{FAAR-RLT:} \quad \text{Maximize} \sum_{p \in \Pi} \sum_{h \in H} f_{ph} \pi_{ph} - \sum_{j \in L} \sum_{k \in K} \sum_{a \in \delta_j^- \cap A^k} (c_{jk} - r_a) x_a \tag{4.40}$$

subject to:

$$\sum_{a \in \delta_j^-} x_a = 1, \qquad \forall j \in L, \tag{4.41}$$

$$\sum_{a \in \delta_j^- \cap A^k} x_a = \sum_{a \in \delta_j^+ \cap A^k} x_a, \qquad \forall j \in L, k \in K, \tag{4.42}$$

$$\sum_{a \in \delta_j^+ \cap A^k} \omega_a^k = t_j \sum_{a \in \delta_j^- \cap A^k} x_a + \sum_{a \in \delta_j^- \cap A_{NM}^k} \omega_a^k, \qquad \forall j \in L, k \in K, \tag{4.43}$$

$$t_a\text{-}x_a \leq \omega_a^k \leq b_a^{t^k} x_a, \qquad \forall a \in A^k, k \in K, \tag{4.44}$$

$$\sum_{a \in \delta_j^+ \cap A^k} \eta_a^k = \sum_{a \in \delta_j^- \cap A^k} x_a + \sum_{a \in \delta_j^- \cap A_{NM}^k} \eta_a^k, \qquad \forall j \in L, k \in K, \tag{4.45}$$

$$x_a \leq \eta_a^k \leq b_a^{to^k} x_a, \qquad \forall a \in A^k, k \in K, \tag{4.46}$$

$$\sum_{a \in \delta_j^+ \cap A^k} \lambda_a^k = \sum_{a \in \delta_j^- \cap A^k} x_a + \sum_{a \in \delta_j^- \cap A_6^k} x_a - \sum_{a \in \delta_j^- \cap A_3^k} x_a + \sum_{a \in \delta_j^- \cap A_{NM}^k} \lambda_a^k, \qquad \forall j \in L, k \in K, \tag{4.47}$$

$$x_a \leq \lambda_a^k \leq b_a^{d^k} x_a, \qquad \forall a \in A^k, k \in K, \tag{4.48}$$

$$\sum_{a \in A_2^k \cup A_4^k} x_a + 2 \sum_{a \in A_5^k \cup A_6^k} x_a + \sum_{j \in L_w} \sum_{a \in \delta_j^- \cap A^k} x_a \leq NA^k, \qquad \forall k \in K, \tag{4.49}$$

$$\sum_{p \in \Pi_j} \pi_{ph} \leq \sum_{k \in K} \sum_{a \in \delta_j^- \cap A^k} \tilde{\gamma}_{jkh} x_a, \qquad \forall j \in L, h \in H, \tag{4.50}$$

$$0 \leq \pi_{ph} \leq \tilde{\mu}_{ph}, \ \text{integer}, \qquad \forall p \in \Pi, h \in H, \tag{4.51}$$

$$x \ \text{binary}, \tag{4.52}$$

where we have used (4.23)-(4.27) to rewrite the objective function and Constraints (4.3)-(4.5), (4.14), and (4.21), and where we have rewritten (4.36) and (4.37) of Proposition 1 in more compact form as (4.43) and (4.44) , respectively, with (4.45)-(4.48) being a similar representation derived for (4.8)-(4.13) and (4.16)-(4.17).

A Lower Bound on the Number of Required Aircraft

In this section, we briefly introduce a valid inequality on the total number of aircraft. Given the flight schedule, suppose that we extend the duration of all flights by the minimum turn-time among all fleet types (denoted by $\tau_{\min}$), and lay out these resulting flight durations on a daily time-line. Then the number of aircraft required must be equal to or exceed the maximum number of overlapping flight intervals at any point in time. Denoting this value by Γ, it follows that

$$\sum_{k \in K} \Big(\sum_{a \in A_2^k \cup A_4^k} x_a + 2 \sum_{a \in A_5^k \cup A_6^k} x_a + \sum_{j \in L_w} \sum_{a \in \delta_j^- \cap A^k} x_a \Big) \geq \Gamma. \tag{4.53}$$

4.2.2 Further Integration with Crew Pairing

We next propose a further integrated airline operational model by extending Model FAAR-RLT to include crew pairing decisions. As mentioned before, crew pairings account for a significant proportion of airline operational costs, and are intimately intertwined with fleet assignment and aircraft routing decisions. Because it is generally difficult to handle these decisions directly by using path-based restrictions along with the accompanying nonlinear crew cost components in a polynomially-sized, but large-scale, modeling framework, we implicitly generate sets of feasible crew pairings and adopt the traditional set partitioning formulation to incorporate crew pairing aspects within Model FAAR-RLT. Toward this end, consider the following additional notation:

F: set of aircraft families, indexed by f.

K^f: set of types of aircraft that belong to family f. (Note that $K = \cup_{f \in F} K^f$.)

P^f: set of admissible crew pairings that can serve flights covered by aircraft of family f.

S^f: set of short connections, indexed by σ, which can be served by aircraft of family f.

$\mathcal{A}_\sigma^f$: set of arcs $a \in \cup_{k \in K^f} A^k$ between the associated pair of flights that correspond to the short connection $\sigma \in S^f$.

e^f_{jp}: binary indicator that equals 1 if flight leg j is covered by pairing $p \in P^f$ of aircraft family f, and 0 otherwise.

$s^f_{\sigma p}$: binary indicator that equals 1 if short connection $\sigma \in S^f$ is covered by pairing $p \in P^f$ of aircraft family f, and 0 otherwise.

w^f_p: cost associated with pairing $p \in P^f$ of aircraft family f.

Furthermore, we introduce the following additional binary decision variables pertaining to pairings:

z^f_p: binary variable that equals 1 if pairing $p \in P^f$ of aircraft family f is selected, and 0 otherwise.

The resulting integrated fleet assignment, aircraft routing, and crew pairing model (**FRC**) can then be formulated as follows:

$$\textbf{FRC:} \qquad \text{Maximize} \quad \sum_{p\in\Pi}\sum_{h\in H} f_{ph}\pi_{ph} - \sum_{j\in L}\sum_{k\in K}\sum_{a\in\delta_j^-\cap A^k}(c_{jk}-r_a)x_a - \sum_{f\in F}\sum_{p\in P^f} w^f_p z^f_p \tag{4.54}$$

$$\text{subject to:} \quad (4.41) - (4.52) \text{ along with:}$$

$$\sum_{p\in P^f} e^f_{jp} z^f_p = \sum_{k\in K^f}\sum_{a\in\delta_j^-\cap A^k} x_a, \qquad \forall j \in L, f \in F, \tag{4.55}$$

$$\sum_{p\in P^f} s^f_{\sigma p} z^f_p \leq \sum_{a\in\mathcal{A}^f_\sigma} x_a, \qquad \forall \sigma \in S^f, f \in F, \tag{4.56}$$

$$z \text{ binary.} \tag{4.57}$$

The objective function (4.54) incorporates an additional crew pairing cost term within (4.40). Constraint (4.55) requires that each flight is served by a crew that is eligible for the specific aircraft family assigned to this flight, where the right-hand side (RHS) of (4.55) equals $\sum_{k\in K^f} y_{jk}$. Constraint (4.56) imposes restrictions on permissible short connections, i.e., a crew pairing pertaining to aircraft family $f \in F$ can cover a short connection $\sigma \in S^f$ only if the two consecutive flights associated with the short connection σ are served by an aircraft from the same family f. Finally, Constraint (4.57) represents the logical binary restrictions on the crew pairing decision variables.

4.3. Solution Methodology

US airline companies usually maintain a daily schedule of hundreds of domestic flights with a variety of large sized fleets. For example, *Delta Airlines* (excluding *Delta Connection*) operated more than 2000 flights with 700 aircraft of different types on a daily basis in 2011. Moreover, a typical airline that uses a hub-and-spoke flight network can hold thousands of itineraries in its profile, and millions of feasible aircraft rotations as well as crew pairings. For such large-sized airlines, our modeling framework can focus on a relatively smaller fleet of wide-body aircraft, along with major associated airports and a related manageable set of principal itineraries in order to assure tractability. On the other hand, there also exist several small sized airlines that maintain a moderate fleet of aircraft and operate only a few flights in their network, for which a full-scale model is practically implementable. Even so, the integrated model yet inherits a huge number of variables and constraints that preclude a direct solution using off-the-shelf software. It is therefore imperative to design a specialized decomposition-based solution methodology, which motivates us to propose a sequential-fixing approach that uses Benders decomposition for solving the integrated Model FRC. In particular, the constraints (4.50)-(4.51) pertaining to itinerary-based demands, and the crew pairing restrictions (4.55)-(4.57), are handled in separate subproblems within the Benders decomposition framework, where the generated cuts therefrom are appended to the aircraft fleeting and routing master program that is comprised of Constraints (4.41)-(4.49) and (4.52)-(4.53).

To begin with, we first solve Model FAAR-RLT in isolation and then subsequently solve the crew pairing problem with the fixed fleeting and aircraft routing decisions. The net solution obtained from these two sequential stages serves as a benchmark for the upcoming integrated model solution, where the optimal crew pairings at hand are also used to initialize the column generation procedure in the corresponding crew pairing subproblem of the integrated model. In the next phase, we turn our attention to Model FRC and decompose it using a Benders partitioning approach, which leads to an itinerary-based passenger-mix subproblem and a crew pairing subproblem, where the Benders relaxed master program addresses aircraft fleeting and routing decisions. In this process, we generate maximal nondominated Benders cuts as proposed by Sherali and Lunday (2011), and we adopt a set of initial cuts for accelerating the solution procedure as recommended by McDaniel and Devine (1977). For the crew pairing subproblem, we employ a perturbed Lagrangian dual strategy along with a deflected subgradient optimization technique as proposed by Subramanian and Sherali (2008) in order to effectively generate useful columns that avoid stalling of the underlying column generation procedure, and we embed this in a branch-and-price heuristic (see Barnhart et al. (1998b) and Mercier et al. (2005)) while adopting the branch-on-the-follow-on strategy (Ryan and Foster, 1981) for selecting branching variables. The detailed steps of this overall procedure are described next in turn below.

4.3.1 Itinerary-based Passenger-Mix Subproblem

In order to facilitate the application of Benders methodology, we initially relax the integrality restrictions on the π-variables. Accordingly, for each fare-class $h \in H$, this yields the following itinerary-based passenger-mix subproblem (IPM$_h$):

$$\textbf{IPM}_h: \quad \text{Maximize} \quad \sum_{p \in \Pi} f_{ph} \pi_{ph} \tag{4.58}$$

subject to:

$$\sum_{p \in \Pi_j} \pi_{ph} \leq \sum_{k \in K} \sum_{a \in \delta_j^- \cap A^k} \tilde{\gamma}_{jkh} \bar{x}_a, \qquad \forall j \in L, \tag{4.59}$$

$$0 \leq \pi_{ph} \leq \tilde{\mu}_{ph}, \qquad \forall p \in \Pi, \tag{4.60}$$

where $\bar{x}$ is the solution inherited from the relaxed Benders master program.

For each $h \in H$, let $\chi^1_{jh}, \forall j \in L$, and $\chi^2_{ph}, \forall p \in \Pi$, denote the dual variables associated with (4.59) and the upper-bounding constraints in (4.60), respectively, and let X_h denote the set of extreme points of the dual feasible region corresponding to Problem IPM$_h$. Because Problem IPM$_h$ has complete recourse, i.e., it is feasible and bounded for any given (nonnegative) solution $\bar{x}$, the optimal value function ϕ_h for Problem IPM$_h$ can therefore be characterized by the following set of Benders optimality cuts:

$$\phi_h \leq \sum_{j \in L} \sum_{k \in K} \sum_{a \in \delta_j^- \cap A^k} \chi^1_{jh} \tilde{\gamma}_{jkh} x_a + \sum_{p \in \Pi} \chi^2_{ph} \tilde{\mu}_{ph}, \forall (\chi^1_{jh}, \chi^2_{ph}) \in X_h, h \in H. \tag{4.61}$$

4.3.2 Crew Pairing Subproblem

The crew pairing subproblem, which is solved separately for each aircraft family $f \in F$, deals with generating sequences of flights for crews to serve while satisfying a series of FAA-mandated work-rule restrictions. The objective is to find a set of feasible pairings that partitions the given aircraft family's flight network at a minimum cost. Specifically, for each $f \in F$, we examine the following continuous relaxation of the crew pairing problem:

$$\mathbf{CP}^f: \quad \text{Minimize} \quad \sum_{p \in P^f} w_p^f z_p^f + \beta \sum_{j \in L} q_j^f \tag{4.62}$$

subject to:

$$\sum_{p \in P^f} e_{jp}^f z_p^f + q_j^f = \sum_{k \in K^f} \sum_{a \in \delta_j^- \cap A^k} \bar{x}_a, \qquad \forall j \in L, \tag{4.63}$$

$$\sum_{p \in P^f} s_{\sigma p}^f z_p^f + q_\sigma^f = \sum_{a \in \mathcal{A}_\sigma^f} \bar{x}_a, \qquad \forall \sigma \in S^f, \tag{4.64}$$

$$(z, q) \geq 0, \tag{4.65}$$

where $q_j^f, \forall j \in L$, are artificial variables appended to Constraint (4.55) along with a penalty term in the objective function using a sufficiently large penalty parameter β that substantially exceeds crew pairing costs, and where $q_\sigma^f, \forall \sigma \in S^f$, are slack variables for Constraint (4.56). These artificial variables ensure that Problem CP^f has an optimal solution for any fixed $\bar{x}$ obtained from the master program, and thus enables us to characterize the corresponding optimal value function in the master program via only Benders optimality cuts. Note that the RHS values in (4.63)-(4.64) are 0 or 1 whenever $\bar{x}$ is binary-valued, which accordingly determine the set of flights that need to be covered by the crews trained to serve on aircraft of family f, including permissible short connections between designated consecutive pairs of flights. Also, note that in the sequel when we consider the continuous relaxation of the Benders master program, we might inherit fractional RHS values in Constraints (4.63)-(4.64). In this case, we shall consider all the flights corresponding to positive RHS values in (4.63)-(4.64) within the network for solving the crew pairing subproblem. Observe that this would yet yield valid Benders cuts.

Due to the nonlinear structure of the cost function and the complexity of the pairing restrictions, Problem CP^f is usually solved using column generation where the pricing subproblem is solved using a *multi-label shortest path* or *constrained shortest path* algorithm as first proposed by Desrochers and Soumis (1989). In this procedure, for a given aircraft family, feasible pairings are generated over a time-space network based on the dual variable values obtained from the restricted master program for Problem CP^f. A pairing always starts and ends at the same crew base station, covering a sequence of admissible flights over multiple days. A series of labels are maintained at the nodes of the corresponding path generated within the network to keep track of constraints pertaining to the accumulated time away from the base, the number of duty periods in the pairing, as well as the flying time, the elapsed time, and the number of flight legs that have been served in any duty. The algorithm thus seeks feasible crew pairing columns having negative reduced costs to iteratively augment the master program.

Theoretically, the multi-label shortest path problem is NP-hard, and could potentially enumerate all pairings for the network before termination; however, if at any node, the label values updated along one potential pairing are all better (smaller) than those for another pairing, then we say that the former pairing *dominates* the latter one, where the dominated pairing can be eliminated from the solution pool. This feature greatly helps reduce the solution effort. In addition, we note a recent attempt by AhmadBeygi et al. (2009) to model the crew pairing problem using a polynomially-sized (compact) integer programming (IP) formulation that can be directly solved using off-the-shelf software. A similar formulation styled in the fashion of the aircraft routing model proposed by Haouari et al. (2011a) is possible for this problem, which could be beneficially lifted using an RLT-based partial convexification process (see Sherali et al. (1998)). Such compact formulations could instead be used to solve the subproblem for generating crew pairings. We leave the investigation of this strategy for future research.

The solution to the LP relaxation of the crew pairing problem for each $f \in F$ provides a Benders optimality cut for the master program. In particular, for each $f \in F$, let $\psi^1_{jf}, \forall j \in L$, and $\psi^2_{\sigma f}, \forall \sigma \in S^f$, denote the dual variables associated with Constraints (4.63) and (4.64), respectively, and let Ψ^f denote the set of extreme points of the dual feasible region corresponding to Problem CP^f. Again, because Problem CP^f has complete recourse, the optimal value function ξ_f for Problem CP^f can be characterized within the master program via the following set of Benders optimality cuts:

$$\xi_f \geq \sum_{j \in L} \sum_{k \in K^f} \sum_{a \in \delta^-_j \cap A^k} \psi^1_{jf} x_a + \sum_{\sigma \in S^f} \sum_{a \in \mathcal{A}^f_\sigma} \psi^2_{\sigma f} x_a, \forall (\psi^1_{jf}, \psi^2_{\sigma f}) \in \Psi_f, f \in F. \tag{4.66}$$

Algorithm for the Crew Pairing Subproblem

For each aircraft family, we adopt a corresponding daily flight network to solve the LP relaxation of the crew pairing subproblem via a column generation approach. As described in Section 4.3.2, the restricted master program for this subproblem is initialized with only artificial columns, where each column covers a single flight leg (or short connection), and is associated with a large penalty (or no cost for short connections). Based on the optimal solutions obtained for the LP relaxation, the associated dual variable values are used to generate candidate pairings via a *constrained shortest path* (CSP) algorithm. This process is terminated when either the optimality gap is reduced to within a specified threshold or a predefined number of iterations has been performed. Below, we present details of a customized CSP scheme that we implemented for solving the crew pairing problem.

Constrained Shortest Path Algorithm

For each crew base of a specific family of aircraft, a dummy source station and a dummy sink station are added to the associated flight network. Correspondingly, a dummy flight denoting the start of a pairing from the source to its associated station is designated at the beginning of the day, and likewise, a dummy flight is designated to link the station back to its sink at the end of the daily timeline. Furthermore, crew connection opportunities between the flights (including dummies) are generated in terms of the arc types $A_1, \ldots, A_6$ as described in the aircraft routing problem, where connections from/to the dummy flights are categorized as a new type. Also, the appropriate reduced costs computed based on the solution to the master program are assigned to each flight and each short connection. The objective is to find paths having negative reduced costs from the source nodes to their corresponding sinks while satisfying several restrictions with respect to the current connections, duties, and pairings. If no such path is found, then the current crew pairing subproblem solution is declared optimal.

As delineated in Section 4.1, the duty cost, measured in hours, is expressed as the maximum of three quantities: the guaranteed minimum working hours, the total flying hours, and a certain fraction of the duty period duration. Moreover, the pairing cost is determined by the maximum value of a predefined guaranteed pay, the total of all duty costs, and a specified percentage of the total *time away from base* (TAFB). In addition, the crew pairing rules adopted in this study specify a set of regulations that can be categorized into the following three types:

1. Connection-dependent rules
   - minimum and maximum sit time between flights.
2. Duty-dependent rules
   - maximum flying time between two consecutive rests;
   - maximum elapsed time between two consecutive rests;
   - maximum number of flights between two consecutive rests.
3. Pairing-dependent rules
   - maximum time away from base;
   - maximum number of duties in the pairing.

Due to the nonlinear cost structure and the work-rule regulations that depend on the specific path from the source to each intermediate node within the network, multiple labels are used to track the concerned performance indices. Whenever an outbound flight connection (arc) is selected from a current node, the labels are correspondingly updated using the information of the particular type of connection as well as the current pairing.

More specifically, we maintain the following principal labels:

1. Current date corresponding to the pairing (*date*).
2. Duty-related labels
   - accumulated flying hours within the current duty ($flytime$);
   - accumulated elapsed hours within the current duty (*elaptime*);
   - number of flights served within the current duty ($nbflights$);
   - cost of the current duty (*dutycost*).

3. Pairing-related labels

    - number of duties served in the pairing (*nbduties*);
    - accumulated elapsed hours for the pairing ($tafb$);
    - accumulated dual solution-based cost for the pairing (*dual*);
    - cost of the pairing (*pairingcost*).

Hence, the duty cost and the pairing cost are respectively evaluated as follows:

$$\begin{aligned} \mathit{dutycost} &= \max\{\mathit{fixedpay},\ flytime,\ \mathit{elaptime}\cdot frac\}, \text{ and} \\ \mathit{pairingcost} &= \max\{\mathit{fixedpay}\cdot \mathit{nbduties},\ \sum_{i=1}^{\mathit{nbduties}} \mathit{dutycost}_i,\ tafb\cdot frac\}, \end{aligned}$$

where $fixedpay$ denotes the minimum daily wage for crew members, and $frac$ stands for a preset percentage value. Furthermore, the reduced pairing cost (*reducedpairingcost*) is obtained by subtracting the dual cost from the pairing cost, i.e.,

$$\mathit{reducedpairingcost} = \mathit{pairingcost} - \mathit{dual}.$$

The constrained shortest path algorithm is analogous to the regular label-correcting algorithm, except that there exist multiple labels that need tracking. For each crew base, the search starts from the initial dummy flight, which is introduced into a *scan-eligible* (SE) node list. At every iteration, a node is *extracted* (selected and deleted) from the SE list in a *first-in-first-out* (FIFO) manner, and outgoing arcs incident at this node are subsequently scanned. If the connection is feasible with regard to the above-mentioned work-rules, the incident to-node is appended to the end of the SE list (unless if it already exists within SE), and the labels on this node are created/updated. After the SE list is depleted, we examine all the recorded pairings on the dummy end-flight, and return those having negative reduced costs to the master program, where they are appended along with their true pairing costs.

In this process, for each flight and the connections emanating from it within the network, any succeeding flight must fall into exactly one of the following four categories:

1. start-flight of a pairing;

2. end-flight of a pairing;
3. a flight that starts a new duty;
4. another flight within the current duty.

The labels of a pairing are therefore updated according to the category pertaining to the next flight. To provide a further detailed description, denote *conntime* as the sit-time of the connection, $nextflighttime$ as the duration of the next flight, and *conndual* and *nodedual* as the dual-based cost values on the connection (if it is short) and the next flight, respectively. Consequently, when starting a new pairing, the *date*-label is initialized with 1, while the other labels are initialized as follows:

$$\begin{aligned}&\{flytime, elaptime, nbflights, nbduties, tafb, dual\} \equiv\\&\qquad\{nextflighttime, nextflighttime, 1, 1, nextflighttime, conndual + nodedual\}.\end{aligned}$$

The duty cost is also updated using the corresponding labels. In contrast, when an end-flight is reached, the pairing is completed with the same labels as for the previous flight.

Moreover, when a connection only violates duty-dependent restrictions, a new duty must be initialized with the following duty-related labels:

$$\{flytime, elaptime, nbflights\} \equiv \{nextflighttime, nextflighttime, 1\},$$

and the pairing-related labels are updated as follows:

$$\begin{aligned}&\{nbduties, tafb, dual\} \equiv\\&\qquad\{nbduties + 1, tafb + conntime + nextflighttime, dual + conndual + nodedual\}.\end{aligned}$$

Note that the cost for the completed duty is evaluated before updating the above labels, and is then used in the calculation of the pairing cost.

When the connection is feasible with respect to all rules, another flight leg is added to the current duty, and the labels are updated as follows:

$$\begin{aligned}&\{flytime, elaptime, nbflights, nbduties, tafb, dual\} \equiv\\&\qquad\{flytime + nextflighttime, elaptime + conntime + nextflighttime, nbflights + 1,\end{aligned}$$

$$nbduties, tafb + conntime + nextflighttime, dual + conndual + nodedual\},$$

and the duty cost is updated accordingly (the pairing cost remains unchanged). Finally, we note that the label *date* is updated whenever an A$_1$- or A$_2$-type connection is selected.

In addition, we also keep track of the flights served within each pairing for the purpose of comparing and eliminating redundant pairings. Since we are considering a daily self-repeated network, there can exist more than one sequence for serving a given set of flights from beginning to end. Moreover, a column in the master program essentially represents a set of flights regardless of the sequence. Therefore, in order to maintain a set of unique columns, only the pairing having the lowest associated cost is considered among all possible sequence permutations, i.e., if two pairings involve the same set of flights but in different orders, only the one having the smaller reduced cost is retained.

Due to the existence of multiple labels, a node is likely to record more than one pairing among its associated labels, where only *nondominated* pairings are maintained (we say that a pairing *dominates* another if it has a lower reduced cost and more flexibility in terms of incorporating successive flights). Specifically, for each flight pertaining to the same day, the pairing having labels with a higher reduced cost as well as higher values of TAFB and the number of duties would in the end yield no better results, and thus can be eliminated (i.e., it is dominated). Hence, all existing pairings recorded for this particular flight are examined so that only nondominated ones remain active. In addition, we note that since we consider a flight schedule that repeats daily, we seek a set of pairings that covers this daily flight schedule, where *posteriori*, as many crews will cyclically adopt the same generated pairing as there are days/duties in the pairing. Hence, we enforce that a flight can be involved in a pairing at most once.

Solving the Crew Pairing Subproblem as an IP

In order to eventually obtain an integral solution to the crew pairing subproblem, we adopt a branch-and-price heuristic, where a *depth-first search* (DFS) tree is employed in a recursive fashion. At each node of the DFS tree, the associated LP relaxation is solved using the algorithm presented in Section 4.3.2. The branch is pruned if the resulting solution obtained is integral. Moreover, if the associated objective value of such a solution is smaller than that of the incumbent solution, we update the current upper bound using this objective value and reset the current solution as the new incumbent.

On the other hand, if the solution obtained for the LP relaxation is not integral and the objective value is greater than the current upper bound, then the branch is fathomed. Otherwise, if the non-integral solution yields an objective value that is smaller than the current upper bound, we apply the *branching-on-the-follow-on* rule (Ryan and Foster, 1981) to create two subnodes in the DFS tree for further exploration. More specifically, as shown in Barnhart et al. (1998b), fractional pairing solutions must at least result in a connection that is only partially covered. Hence, we branch on a particular fractional connection, enforcing it on the left branch (i.e., eliminating all other connections from the previous flight and those to the following flight), and forbidding it on the right branch (i.e., deleting it from the network). Furthermore, within the depth-first scheme, the left branch is usually explored prior to the right branch, with the hope that this sequential fixing strategy (yet with the backtracking mechanism in place) can quickly lead to a desirable solution.

A number of alternative strategies can be adopted for selecting the branching connection. For example, a common strategy is to choose the connection having the greatest *fractionality* (i.e., $\arg\max\{\min\{x_a, 1-x_a\} : a \in A\}$, where x_a represents the fractional value of the connection). Another strategy is to select a connection having the greatest fractional value (i.e., $\arg\max\{x_a : x_a \neq 1, a \in A\}$), hoping that a good CP solution can be obtained quickly by first enforcing this connection. We use the latter strategy herein.

The unexplored nodes in the DFS tree are maintained in a *last-in-first-out* (LIFO) manner until all nodes are pruned. Also, the algorithm terminates whenever the percentage optimality gap between the upper and lower bounds falls within a predefined threshold. We note that another aggressive criterion that is sometimes implemented (e.g., in Papadakos (2009)) terminates the search once an integral solution is obtained.

4.3.3 Accelerating the Solution of the Crew Pairing Problem

The column generation approach used for the crew pairing problem often stalls when yet remote from optimality, resulting in dual variable values that help little in generating improving pairings via the subproblem. To avoid such a phenomenon, which is identified as *dual noise* by Subramanian and Sherali (2008), we follow the authors' approach by solving a perturbed Lagrangian dual of the restricted crew pairing master program using a deflected subgradient optimization technique. This process has been shown to derive judicious low-norm dual solutions that generate beneficial columns. For the sake of convenience, we restate the restricted master program for Problem CP^f for any aircraft family $f \in F$ in a generic form as follows:

$$\textbf{CP}: \quad \text{Minimize} \quad \{\omega^T z + \beta e_L^T q : Bz + q = R, (z, q) \geq 0\}, \tag{4.67}$$

where e_L is a vector of $|L|$-ones and $|S^f|$-zeroes (for $f \in F$). Following Subramanian and Sherali (2008), we perturb this problem by accommodating a term $m\alpha$ in the constraints along with an associated quadratic penalty term $\frac{m}{2}\alpha^T\alpha$ in the objective function to obtain the following problem:

$$\textbf{QCP}: \quad \text{Minimize} \quad \omega^T z + \beta e_L^T q + \frac{m}{2}\alpha^T \alpha \tag{4.68}$$

subject to:

$$Bz + q + m\alpha = R, \tag{4.69}$$

$$(z, q) \geq 0, \ \alpha \text{ unrestricted}, \tag{4.70}$$

where m is a small perturbation parameter (typically, $m = 10^{-7}$), and α is a perturbation vector. By Dorn's duality (Dorn, 1960), the dual to Problem QCP inherits a term $-\frac{m}{2}\|\psi\|^2$ in the objective function (also, see (4.73) below), where as before, ψ denotes the dual variables associated with the equality constraints in QCP, thus encouraging the generation of low-norm dual solutions. This provides a dual stabilization effect for the column generation procedure as discussed in De Merle et al. (1999). Accordingly, consider the following Lagrangian dual for Problem QCP:

$$\textbf{LD}: \max_{\psi \text{ unrestricted}} \theta(\psi), \tag{4.71}$$

where $\theta(\psi)$ is given by the optimal value to the following Lagrangian subproblem, where Z represents implied interval bounds of $[0, 1]$ on the (z, q)-variables, imposed to ensure a finite value for $\theta(\cdot)$:

$$\textbf{LS}: \quad \theta(\psi) = \min_{\substack{(z,q) \in Z, \\ \alpha \text{ unrestricted}}} \{\omega^T z + \beta e_L^T q + \frac{m}{2}\alpha^T\alpha + \psi^T(R - Bz - q - m\alpha)\}. \tag{4.72}$$

Note that the necessary and sufficient first-order optimality condition with respect to α in (4.72) yields $\alpha = \psi$, whence the Lagrangian subproblem reduces to the following, where the fixed term $-\frac{m}{2}\|\psi\|^2$ induces low-norm dual solutions among alternative optimal dual solutions as noted above:

$$\mathbf{LS}: \theta(\psi) = \psi^T R - \frac{m}{2}\|\psi\|^2 + \min_{(z,q)\in Z}\{(\omega - B^T\psi)^T z + (\beta e_L - \psi)^T q\}. \tag{4.73}$$

Note that $\theta(\psi)$ is trivially evaluated via (4.73). Hence, given any dual solution ψ_t at iteration t, if (z_t, q_t) evaluates $\theta(\psi_t)$ via (4.73), then a subgradient g_t of $\theta(\cdot)$ at $\psi = \psi_t$ is given by

$$g_t = R - m\psi_t - Bz_t - q_t. \tag{4.74}$$

We adopt the deflected subgradient method (DSG) proposed by Subramanian and Sherali (2008) to solve Problem LD. In each iteration, the search direction is determined by a subgradient of the nondifferentiable function $\theta(\cdot)$ that is appropriately deflected using the information from the previous iteration in order to approach an *implicit target value* of the objective function in the Euclidean norm sense. A step size is prescribed along the search direction based on certain target value-related and step size parameters, where the latter is periodically halved whenever the incumbent objective value fails to improve by a threshold level over a specified number of iterations. The procedure is terminated whenever either the l_∞ norm of the generated direction is sufficiently small or if the imposed maximum iteration limit is reached. The resulting dual solution is then used to generate additional columns as necessary. Once the column generation process terminates, the final dual solution ψ is used to generate the Benders cut (4.66) as necessary. The DSG algorithm is summarized below.

Algorithm DSG:

Initialization. At iteration $t = 1$, set the incumbent dual solution $\bar{\psi} \equiv 0$, and compute $\bar{\theta} \equiv \theta(\bar{\psi})$. Furthermore, set the target-related parameter $\varepsilon_T = 1$, and initialize $\psi_1 = \bar{\psi}, \theta_1 = \bar{\theta}$, and use the search direction $d_1 = g_1$ as given by (4.74). Also, set $\hat{\theta} \equiv \bar{\theta}, \kappa_{\min} = 10$, and let $\kappa = \kappa_{\min} + 1$.

Main Iterative Step. Update the dual solution as follows:

$$\psi_{t+1} = \begin{cases} \hat{\psi} + 2\zeta\frac{\|\hat{\psi}\|}{\|d_t\|}d_t, & \text{if } \|\hat{\psi}\| \geq \varepsilon_T\|d_t\|, \\ \hat{\psi} + 2\zeta\varepsilon_T d_t, & \text{otherwise,} \end{cases}$$

where ζ denotes a suitable step-length parameter with $0 < \zeta_{\min} \leq \zeta \leq 1$ (we use $\zeta_{\min} = 10^{-4}$), and where

$$\hat{\psi} = \begin{cases} \psi_t, \text{ if } \|\psi_t\| < \|\bar{\psi}\|, \\ \bar{\psi}, \text{ otherwise.} \end{cases}$$

Compute $\theta_{t+1} \equiv \theta(\psi_{t+1})$ and g_{t+1} via (4.73) and (4.74), respectively. If $\theta_{t+1} > \bar{\theta}$, then update the incumbent solution $\bar{\psi} \leftarrow \psi_{t+1}$ and $\bar{\theta} \leftarrow \theta_{t+1}$. Increment $t \leftarrow t+1$.

Parameter Update. If $t = \kappa$, do:

(a). If $\bar{\theta} - \hat{\theta} < \varepsilon_\theta$ (we use $\varepsilon_\theta = 0.1$), then set $\zeta \leftarrow \max\{\zeta/2, \zeta_{\min}\}$; else, retain ζ.

(b). Reset $\hat{\theta} = \bar{\theta}$ and increment $\kappa = t + \kappa_{\min}$.

New Direction. Compute the next deflected subgradient-based direction as $d_{t+1} = \zeta g_{t+1} + (1-\zeta)d_t$.

Termination Check. If $\max_{i=1,\ldots,|L|+|S^f|} |d_{ti}| < \varepsilon_d$ (we use $\varepsilon_d = 0.05$), or a specified maximum number of iterations ($t_{\max}$) is reached (we use $t_{\max} = 2000$), terminate this procedure and exit with the incumbent dual solution $\bar{\psi}$ with objective value $\bar{\theta}$. Else, repeat the Main Iterative Step.

4.3.4 Accelerating the Benders Decomposition Procedure

The Benders relaxed master program (RMP) is comprised of the aircraft fleeting and routing decisions and constraints, along with Benders cuts obtained from the passenger-mix and the crew pairing subproblems, and is given as follows:

RMP: $$\text{Maximize} \sum_{h \in H} \phi_h - \sum_{j \in L} \sum_{k \in K} \sum_{a \in \delta_j^- \cap A^k} (c_{jk} - r_a) x_a - \sum_{f \in F} \xi_f \quad (4.75)$$

subject to:

$$(4.41) - (4.49), (4.52), \text{ along with:}$$

$$\phi_h \leq \sum_{j \in L} \sum_{k \in K} \sum_{a \in \delta_j^- \cap A^k} \chi_{jh}^1 \tilde{\gamma}_{jkh} x_a + \sum_{p \in \Pi} \chi_{ph}^2 \tilde{\mu}_{ph}, \qquad \forall (\chi_{jh}^1, \chi_{ph}^2) \in X_h^r, h \in H, \quad (4.76)$$

$$\xi_f \geq \sum_{j\in L}\sum_{k\in K^f}\sum_{a\in\delta_j^-\cap A^k}\psi_{jf}^1 x_a + \sum_{\sigma\in S^f}\sum_{a\in\mathcal{A}_\sigma^f}\psi_{\sigma f}^2 x_a, \qquad \forall(\psi_{jf}^1, \psi_{\sigma f}^2) \in \Psi_f^r, f \in F, \tag{4.77}$$

where $X_h^r \subseteq conv(X_h), \forall h \in H$, and $\Psi_f^r \subseteq conv(\Psi_f), \forall f \in F$, are appropriate restricted sets of dual solutions that correspond to the Benders optimality cuts generated thus far, and where $conv(\cdot)$ denotes the convex hull operation.

We employ the implementation scheme advocated by McDaniel and Devine (1977) to accelerate the solution procedure by initially relaxing the integrality restrictions within the master program and thus solving the continuous relaxation of Problem FRC via the Benders decomposition method. If the lower and upper bounds obtained respectively from the relaxed master program and the Benders subproblem are sufficiently close, or some specified limit of iterations is reached, then we re-enforce the integrality restrictions on the x-variables as noted below, and continue the solution of the resulting mixed-integer program using the implementation of Benders procedure as recommended by Geoffrion and Graves (1974). In this process, rather than solve each current relaxed master program to optimality as a mixed-integer program (MIP), we essentially solve just the full master program itself (denoted by MP) as an MIP, which is equivalent to solving the original problem FRC. This is accomplished by solving Problem MP using a branch-and-cut (B&C) approach, where Benders cuts are progressively generated as-and-when needed. Specifically, in this B&C algorithm, upper bounds are computed by solving the LP relaxation of the current RMP. Now, suppose that an integer-feasible solution is detected to RMP whose objective value is greater than the current lower bound (incumbent) value plus a prescribed threshold $\varepsilon_{RMP} \geq 0$. In this case, we pause and solve the Benders passenger-mix and crew pairing subproblems to evaluate the true objective function value of this solution in Problem MP (and therefore in Problem FRC) in order to update the incumbent solution to MP and generate new Benders cuts as necessary, and we then continue the B&C algorithm. When the B&C algorithm discovers that no such integer-feasible solution exists, we terminate the overall algorithm with the indication that the incumbent solution is ε_{RMP}-optimal to the original problem.

Furthermore, in order to accelerate the convergence of the Benders decomposition method, Magnanti and Wong (1981) presented a seminal work for generating *nondominated* or *Pareto-optimal* Benders cuts using a core point that is selected within the relative interior of the convex hull of the primal feasible region defining the master program. Such nondominated cuts are particularly useful when the primal subproblem is highly degenerate, which induces a variety of possible Benders cuts at any iteration based on alternative optimal dual solutions. Magnanti and Wong proposed to solve a secondary subproblem to appropriately select among the alternative optimal solutions to generate Pareto-optimal cuts. Because Problem CP is known to be highly degenerate, Papadakos (2009) adopted this technique to generate nondominated cuts using a heuristic for finding core points. However, as demonstrated by Mercier and Soumis (2007) and by Sherali and Lunday (2011), although the Magnanti-Wong method reduces the total number of generated cuts, it may not lead to a net advantage in computational effort because it requires solving an additional linear program at each iteration. As an alternative mechanism, Sherali and Lunday proposed a perturbation scheme for the subproblem that automatically generates maximal-nondominated Benders cuts via a single subproblem optimization step, which is also convenient for implementation. Following this strategy for Problem IPM$_h, \forall h \in H$, we perturb the RHS of Constraints (4.59) and (4.60) as specified below:

$$\sum_{p\in\Pi_j} \pi_{ph} \leq \sum_{k\in K} \sum_{a\in\delta_j^-\cap A^k} \tilde{\gamma}_{jkh}(\bar{x}_a + \epsilon\hat{x}_a), \qquad \forall j \in L, \tag{4.78}$$

$$0 \leq \pi_{ph} \leq (1+\epsilon)\tilde{\mu}_{ph}, \qquad \forall p \in \Pi, \tag{4.79}$$

where $\hat{x}$ is a predefined positive weight vector (we use $\hat{x}_a = 1, \forall a \in A$), and where ϵ is a perturbation coefficient (we use $\epsilon = 10^{-6}$). Note that the dual solution generated for the perturbed subproblem is also feasible to the original dual subproblem and so the resulting Benders cut remains valid. However, the dual objective function of the perturbed subproblem inherits a corresponding perturbation term that guides the selection of a dual solution among (near-) alternative optimal solutions in order to derive maximal nondominated Benders cuts.

Likewise, for Problem CP$^f, \forall f \in F$, we rewrite Constraints (4.63) and (4.64) with appropriate perturbations as:

$$\sum_{p\in P^f} e^f_{jp} z^f_p + q^f_j = \sum_{k\in K^f} \sum_{a\in\delta_j^-\cap A^k} (\bar{x}_a + \epsilon\hat{x}_a), \qquad \forall j \in L, \tag{4.80}$$

$$\sum_{p\in P^f} s^f_{\sigma p} z^f_p + q^f_\sigma = \sum_{a\in\mathcal{A}^f_\sigma} (\bar{x}_a + \epsilon\hat{x}_a), \qquad \forall \sigma \in S^f, \tag{4.81}$$

and we proceed with the solution approach as described above, where the resulting dual solutions are used to derive the Benders cuts (4.76) and (4.77) in the same form as before (i.e., without the perturbation terms).

In addition, we adopted a tabu-type strategy to help alleviate an observed stalling phenomenon for the relaxed master program by suppressing repetitive Benders subproblems. Due to the inherent symmetry in the model, distinct $\bar{x}$-solutions could lead to identical Benders subproblems by providing the same RHS values. Specifically, since Constraints (4.58)-(4.60) in Model IPM are essentially determined by the fleet assignment decisions $\bar{y}_{jk} \equiv \sum_{a \in \delta_j^- \cap A^k} \bar{x}_a$, different $\bar{x}$-values could result in the same fleet assignment, thus creating the same subproblem. Therefore, we keep track of the distinct fleeting decisions using a tabu list, and call the subproblem only when the $\bar{x}$-values indicate a new fleet assignment pattern. Likewise, Subproblem CP (given by (4.62)-(4.65)) is invoked only when any flights are assigned to a different aircraft family, or if the involved short connections are changed.

The overall solution procedure is summarized below in two phases.

Algorithm A:

Phase I: **Sequential Optimization**:

(a) Solve Problem FAAR-RLT as an MIP to obtain fleet assignment and passenger-mix decisions $(\tilde{x}, \tilde{\pi})$.

(b) Sequentially, solve the crew pairing problem given by the Constraints (4.54)-(4.57) of Problem FRC with (x, π) fixed at $(\tilde{x}, \tilde{\pi})$. In this process as described in Sections 4.3.2 and 4.3.2, begin by solving the continuous relaxation of Problem $\text{CP}^f, \forall f \in F$, via column generation, and then apply a suitable branch-and-price-based heuristic (denoted by **BPH**) to obtain an integer-feasible solution $\tilde{z}$. Let $\tilde{\nu}$ be the objective value of Problem FRC corresponding to the solution $(\tilde{x}, \tilde{\pi}, \tilde{z})$, and let $\tilde{P}^f$ be the set of pairings thus generated for each $f \in F$.

Phase II: **Integrated Optimization**:

(a) Begin by solving the continuous relaxation of Problem FRC via Benders decomposition to generate an initial set of Benders cuts as per McDaniel and Devine (1977). In this process, use the Phase I solution $(\tilde{x}, \tilde{\pi})$ to generate initial Benders cuts from the passenger-mix and crew pairing subproblems, and adopt the pairings in $\tilde{P}^f$ (as appropriate, depending on the fleet assignment solution passed to the subproblem) to initialize the set of columns within the subproblem $\text{CP}^f, \forall f \in F$.

(b) Let $\hat{\mathrm{FRC}}$ denote Problem FRC with only the x-variables restricted to be integer-valued. Using the initial set of Benders cuts generated at Step II(a), solve Problem $\hat{\mathrm{FRC}}$ via Benders decomposition while applying the approach of Geoffrion and Graves (1974) as described in Section 4.3.4, where the master program is solved just once as an overall integer program, with additional maximal nondominated Benders cuts (as per Sherali and Lunday (2011)) being iteratively generated in the spirit of a branch-and-cut algorithm. Moreover, in this process, when using column generation to solve the crew pairing problem, apply Algorithm DSG of Subramanian and Sherali (2008) to help generate judicious columns. Let x^* be the resulting fleet assignment and routing solution thus obtained. Solve the passenger-mix subproblems $\mathrm{IPM}_h, \forall h \in H$, while enforcing integrality on the π-variables to derive a corresponding passenger-mix solution π^*. Next, solve the crew pairing problem using BPH as in Step I(b) with (x, π) fixed at (x^*, π^*). Let z^* denote the resulting crew pairing decisions. Prescribe the (principal) solution (x^*, π^*, z^*) for Problem FRC, with corresponding objective value ν^*. Also, compute $\frac{(\nu^* - \tilde{\nu})}{\tilde{\nu}} \cdot 100\%$ as the percentage improvement in profits for the integrated solution over the sequential solution.

Note: In our implementation, the LP relaxations in the branch-and-cut algorithm in Step I(a), as well as the LP relaxations of the master programs in Steps I(b), II(a), and II(b) were solved using the Barrier-option of CPLEX.

4.4. Computational Results

In this section, we present numerical results using a series of test instances based on historical data obtained from *United Airlines*, a US legacy airline carrier. For each of the test scenarios, the two-phase procedure (Algorithm A) was adopted as described in the previous section, i.e., we solved each of the test scenarios using the sequential approach and the proposed integrated FRC model, where in the former, the partially integrated Model FAAR-RLT is solved separately from the conventional crew pairing model. This solution framework was implemented using C++ along with the off-the-shelf MIP solver CPLEX 12.4 with default settings (except as noted). All tests were performed on a workstation having dual quad-core Intel Xeon 2.4G CPUs with 24GB RAM and running a 64-bit Linux system (Fedora 16).

4.4.1 Data Description

The instances used for our experiments were derived from the North American flight schedule of United Airlines based on its most frequently served airports. Since the network has dense flights between such hub and major stations that attract many travelers, there exist large numbers of connection opportunities for aircraft as well as crew members. Additionally, we note that arcs of types A_5 and A_6 (arrivals that connect to third-day departures) were omitted when generating connections, since they are unfavorable from the perspective of an efficient schedule and therefore rarely occur in practice.

To cover the flight network, we principally considered mid- and large-sized jets, namely, the Airbus 320 series (including fleets of Airbus 319 and 320), the Boeing 757 series (including fleets of Boeing 757-200 and 757-300), and a fleet of the Boeing 777-200 series used for domestic service. Based on the hourly fuel consumption rate for each fleet type, the flight operational cost was estimated using the flight duration, which was further incremented by a fixed percentage (5%) to account for any unplanned detours.

Fleets that are customized for domestic flights usually have three cabin classes, i.e., the business class, the economy-plus class that provides extra leg-room, and the regular economy class. For the sake of simplicity, we assumed that travel demands for each O-D pair are categorized, and therefore priced, according to these actual cabin classes, although in practice there exists a variety of fare-classes for each cabin. Moreover, for each fleet type, the number of seats in each class was taken as fixed and independent (i.e., not nested). Whenever the demand for a particular flight-cabin on any flight leg exceeds the corresponding capacity on the assigned aircraft, we assume that a certain proportion (20%) of the spilled passengers are lost; this was accordingly used to account for the opportunity cost factor within (4.1).

Moreover, each fleet type has its own maintenance requirements between two consecutive A-checks, which govern the specified maximum flying hours ($t_{\max}$), the maximum number of days ($d_{\max}$), and the maximum number of take-offs ($to_{\max}$). Our data sets and implementation focused on monitoring and enforcing the maximum flying days requirement, where the restriction on the consecutive flying minutes was additionally automatically guaranteed by assuming that $1440 \cdot d_{\max} \leq t_{\max}$.

As noted before, crew members are classified according to the aircraft family that they are eligible to serve. Crew costs are measured in hours, and are determined by a minimum guaranteed pay (assumed to be based on four hours a day), the actual flying time, and a proportion of the total time away from base (assumed to be 80%). Furthermore, in order to simplify crew costs in a compatible manner with other expenditures, we used a fixed rate to estimate the total hourly salary of pilots and flight attendants serving a pairing. Also, we designated eight particular stations as crew bases, so that crew pairings must start and end at one of these stations.

Seven instances, as presented in Table 4.1, were generated for testing the proposed solution methodology. These instances consider progressively increasing sized networks that include major airports in the continental US, where in particular, Instance 3 focuses on flights between five key hubs. Here, we only consider the major airports in the US that involve several daily flights because (i) they deal with a significant portion of the daily travel demand, and (ii) for other smaller spoke airports, the flights are usually less frequent and are well scheduled, and can thus be preprocessed appropriately. The number of connections displayed in Table 4.1 reflects the number of binary variables for each designated fleet type. Moreover, since we assume that only the Airbus 320 family has a minimum turn-time that is shorter than the specified sit-time (45 min), short connections are considered only in the connection networks for the Airbus 320 series aircraft types. Although a typical full-scale daily domestic network has many more flight legs than our largest sized data instance, our purpose here is to mainly focus on suitable dense sub-networks that represent principal connection opportunities. Hence, these instances are both significant and challenging.

4.4.2 Results for the Sequential Approach

We first present computational results for Phase I of Algorithm A, which involves solving the fleet assignment and aircraft routing problem (FAAR) and then sequentially solving the crew pairing problem (CP) separately for each aircraft family based on the aircraft assignments obtained from the solution to Model FAAR-RLT. Table 4.2 reports the results obtained for Model FAAR-RLT, and Table 4.3 provides the results for the subsequent solution of Model CP.

From the numerical perspective, the results demonstrate the efficacy of the partially inte-

Table 4.1: Description of Test Instances

	Flights	Sta.	Total Connections			Short Connections	# of Aircraft			Total Itineraries
			A320	B757	B777		A320	B757	B777	
Test 1	128	17	4734	4655	4313	33	17	17	1	1642
Test 2	154	10	8687	8457	8133	49	32	19	2	1315
Test 3	205	5	10609	10445	10094	67	42	26	2	2319
Test 4	246	19	18688	18282	17552	102	49	31	2	3516
Test 5	354	25	23781	23369	22209	192	71	44	3	6124
Test 6	440	24	38622	37861	36127	231	88	55	4	7976
Test 7	522	23	44435	43629	41645	274	104	64	5	9075

grated Model FAAR-RLT, where all data instances were solved effectively within reasonable computational times. The solution time required for the largest data instance was less than five hours, while the other instances were solved within several minutes.

We also compared the lower bound on the minimum number of required aircraft for each family, as computed in Section 4.2.1, with the actual number given by the solution. Apparently, more aircraft than necessary were needed in order to run the network efficiently and to maintain a cost-effective schedule. Furthermore, among the three aircraft families, because the Airbus 320 has the lowest hourly operating cost, it was utilized to the fullest extent for all test instances. Correspondingly, this family covered a large proportion of flights in the network, ranging from about 50% to more than 75% of the total number of flights. On the other hand, for the given test data, the Boeing 757 aircraft were used to a lesser extent, and the Boeing 777 aircraft were seldom deployed.

Given the aircraft assignments to the flight legs as determined by the solution to Model FAAR-RLT, the flight network gets correspondingly partitioned into sub-networks according to aircraft families. For each family, the crew pairing model was then solved using the branch-and-price heuristic **BPH**. Moreover, in order to curtail the solution effort, the solution to CP was terminated upon finding the first integer-feasible solution. This strategy was well justified in our computational experience since the resultant gap between the final objective value and that from the LP relaxation turned out to be sufficiently small (usually less than

Table 4.2: Fleet Assignment and Aircraft Routing Results

FAAR-RLT		Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Comp. Time (*min.*)		1.80	4.89	3.37	22.26	36.90	83.64	260.42
Benders Cuts		2	1	1	2	2	1	2
LB on Required Aircraft (as per Section 4.2.1)		30	33	56	57	88	95	117
Aircraft Used	A320	17	32	42	49	71	88	104
	B757	17	6	24	31	44	55	64
	B777	1	0	0	2	3	4	5
Flights Covered by	A320	68	140	165	150	216	278	349
	B757	58	14	41	90	128	150	161
	B777	2	0	0	6	10	12	12

0.5%).

The results presented in Table 4.3 show that the branch-and-price heuristic solved the test instances within a few minutes to several hours depending on the structure of the problem, which influences the number of feasible crew pairings generated at the root node by the constrained shortest path algorithm, as displayed in the table. This is evident for Instance 3 that concerns a network involving key hubs, for which there exist abundant connection opportunities at each involved station, leading to even more pairing opportunities for crew members. Furthermore, the results show that only a few short connection opportunities are adopted.

In order to assess the quality of the crew pairing results, the CP cost (converted to hours) was also benchmarked against the total flight duration for the given instance. Unsurprisingly, the network involving only the key hubs has a smaller cost-to-flight duration ratio since (i) at each station, there exist many connection opportunities that involve short sit-times, and (ii) these key hubs are mostly crew bases, which helps save on overnight costs. On the other hand, for a typical hub-and-spoke network, the actual CP cost per flight hour is relatively higher, but could be reduced by recruiting crews from an extended list of stations.

Table 4.3: Crew Pairing Results

CP		Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Comp. Time (*min.*)		0.11	5.79	140.30	8.97	12.91	192.00	985.17
Pairings Generated	A320	728	3547	5697	3011	4355	6102	8516
	B757	553	52	434	818	937	2233	1694
	B777	3	0	0	12	28	36	33
# of Short Connections		6	13	11	3	13	9	20
Total Flight Time (*hr.*)		216.6	309.46	659.6	578.82	1006.1	1030.82	1749.8
CP Cost (*work-hr.*)		658.5	583.2	801.88	1155.5	2010.7	2151.49	2557.2
Ratio of CP Cost to Total Flight Time		3.04	1.88	1.22	2.00	2.00	2.09	1.46

4.4.3 Results for the Integrated Fleeting, Routing, and Crew Pairing Approach

We next present numerical results for Phase II of the proposed Algorithm A, which involves solving the integrated Model FRC. Because this solution process repeatedly solves the passenger-mix and crew pairing subproblems as opposed to the sequential approach of Phase I, it consumes a relatively greater computational effort, ranging from less than an hour to more than 10 hours. Yet, a significant improvement is achieved in profits over that from the partial integration in Phase I, and it can be reckoned that the improvement over the conventional sequential approach (which further separates the solution of the fleet assignment and the aircraft routing problems) would only be greater.

Table 4.4 presents the results obtained, where we further provide insights into the relative nature of solutions derived and their net profit potential in comparison against the results obtained from Phase I. As it turns out, Phase II achieves additional profit by way of decreased crew pairing costs, which results from the manner in which the integrated approach effectively adjusts the flight subnetwork for each aircraft family in order to help generate better crew pairings. In particular, the larger-sized jets, i.e., the Boeing 757s and 777s, were assigned to more flights legs so as to facilitate improved crew pairing decisions and thus lower the overall associated costs. Although the aircraft operational costs were subsequently higher, the look-ahead mechanism afforded by the integrated approach compensated for this increase by lowering crew costs, thus resulting in an overall cost reduction. The additional resulting profits (2.73% on average, which translates to about 43 million dollars per year) demonstrate the effectiveness of the proposed integrated approach, which beneficially overcomes the shortsightedness of the sequential approach.

Table 4.4: Results for the Integrated Approach

FRC		Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Comp. Time for FRC (*min.*)		47.30	102.26	648.15	73.57	234.29	149.57	316.89
Benders Cuts		18	9	23	2	10	2	2
Aircraft Used	A320	17	32	42	48	71	88	104
	B757	17	9	25	31	44	55	64
	B777	1	0	1	2	3	4	5
Flights Covered by	A320	69	138	149	139	213	256	319
	B757	57	16	54	101	131	176	191
	B777	2	0	2	6	10	8	12
Comp. Time for CP (*min.*)		0.23	3.98	9.04	4.40	9.61	76.91	1208.30
CP Cost (*work-hr.*)		591.7	582.2	778.45	1097.4	1882.6	2066.09	2470.5
# of Short Connections		6	10	8	5	12	19	15
CP Cost Reduction w.r.t. Phase I		10.14%	0.17%	2.92%	5.03%	6.37%	3.97%	3.39%
Obj. Value Improvement w.r.t. Phase I		7.25%	2.45%	0.38%	2.43%	3.33%	0.68%	2.57%

4.5. Summary and Conclusions

In this chapter, we have presented an integrated model along with associated solution methodologies for the airline scheduling process. The proposed model incorporates itinerary-based fleet assignment, aircraft routing, and crew pairing within a single framework. We developed a layered flight connection network, where the flights were duplicated for each fleet type, and connections for a particular fleet type were constructed based on the associated maintenance stations and required work-times as well as the minimum turn-time at any station. The model therefore extends the node-arc formulation for the aircraft routing problem proposed in Haouari et al. (2011a) to incorporate fleet assignment decisions, while retaining a compact, polynomial-sized linear MIP. Furthermore, we included within this model additional features of itinerary-based demands along with crew pairing decisions, where the latter therefore simultaneously generate minimum-cost rotations for crew groups according to their eligible aircraft families.

The complex structure of the problem and the large-scale of the model formulation inhibit a direct solution using off-the-shelf software packages even for small-sized data sets. Hence, in order to effectively tackle this integrated Model FRC, we adopted a Benders decomposition solution framework where the fleeting and routing decisions were treated within the master program, and the passenger-mix and crew pairing decisions were handled via separate subproblems. In particular, due to the nonlinear cost structure and various work-rules in the crew pairing subproblem, we further embedded a column generation algorithm within a Lagrangian relaxation scheme, where a deflected subgradient algorithm (Subramanian and Sherali, 2008) was applied to solve the restricted master program, and where eligible pairings were derived using a constrained shortest path algorithm. Moreover, we generated maximal nondominated Benders cuts as per Sherali and Lunday (2011) via the passenger-mix and crew pairing subproblems, and we adopted several other acceleration strategies such as the Benders scheme proposed by McDaniel and Devine (1977) and Geoffrion and Graves (1974) in order to generate an initial set of cuts and to effectively manipulate the master program, repsectively.

To demonstrate the benefits of this integrated modeling approach, we proposed a two-phase solution framework, where in Phase I we solved Model FAAR-RLT that represents a partially integrated itinerary-based fleet assignment and aircraft routing problem, followed by a sequential solution of a crew pairing problem for each sub-network based on the resulting fleet assignment decisions. In Phase II of this framework, we solved the fully integrated Model FRC as described above, using a set of initial cuts as generated during Phase I. Several test instances were derived using historical data obtained from *United Airlines*, a legacy US carrier. The computational results obtained demonstrated the value of adopting an integrated viewpoint in Phase II as opposed to the (partially) sequential approach of Phase I. On average over our set of seven test instances, this achieved an improved profit of about 2.73% (or an estimated 43 million dollars per year), which accrued mainly from a judicious, albeit more expressive, utilization of the available fleet in a manner that significantly decreased the crew pairing costs, thus reducing the overall costs.

Chapter 5

Summary, Conclusions, and Recommendations for Future Research

This dissertation has studied the airline scheduling decision-making process that involves fleet assignment with itinerary-based demands, aircraft routing, and crew pairing, using mathematical optimization models and solution techniques. We began by developing a novel compact node-arc formulation for the daily aircraft maintenance routing problem, based on which we designed an integrated model that further incorporates fleet assignment and crew pairing decisions. These models, along with proposed specialized decomposition-based solution procedure, were tested using real-life data sets obtained from *United Airlines* for the purpose of demonstrating their efficacy and for providing insights to the airline industry.

The model for the aircraft routing problem (ARP) considers a daily repeated flight network for a specific aircraft fleet type, where individual aircraft in this fleet are assigned to each flight leg under a series of mandatory maintenance regulations. The objective function of the ARP maximizes the *through-value*, which is defined as the extra profit gained from providing convenient connection opportunities. In addition, the ARP is sometimes treated as a feasibility problem that finds a set of suitable rotations for aircraft. To address the ARP, we proposed a polynomial-sized formulation that involves three main maintenance indices, i.e., the flying time, the number of days, and the number of take-offs between two consecutive maintenance checks. This novel node-arc formulation, which has nonlinear constraints, was linearized using the Reformulation-Linearization Technique (RLT), and was further enhanced via suitable valid inequalities. In addition, we proposed two root-node strategies, i.e., a partial convexification process and another set of valid inequalities derived from it, in order to improve the representation of the model and thus enhance its solvability. The resulting lifted model was solved using a commercial solver (CPLEX). Five test instances were generated using historical data obtained from *United Airlines*. The computational results demonstrated the utility of our enhanced formulation, where the largest instance with 344 daily flight legs was optimized in about 10 CPU seconds.

Next, we extended this modeling concept by incorporating the other decision-making stages of fleet assignment and crew pairing from the airline scheduling process. Because of the interdependencies between these various stages within traditional sequential solution approaches, when the results from the upstream fleet assignment stage are passed to the downstream stages of aircraft routing and crew pairing as given inputs, this limits the scope of the latter and leads to sub-optimal, or even possibly infeasible, solutions. In addition, the aircraft routing problem is also intertwined with the crew pairing model due to the rules involving short connections for crews. Consequently, a single integrated framework that incorporates all the three concerned stages is beneficial for capturing the interdependencies and thus obtaining improved solutions.

With this motivation, we developed an integrated framework for the airline scheduling problem, where we first extended the proposed aircraft routing model to incorporate fleet assignment decisions by introducing a layered flight network for each fleet type. Moreover, we considered within this model the relatively more practical features of itinerary-based demands and fare-class-based seat capacities in various cabin-classes for each type of aircraft. As a next major step, we incorporated within the model crew pairing decisions, which generates for each aircraft family a set of feasible crew rotations that satisfy a series of mandated FAA-imposed work-rules. This integrated fleeting, routing, and crew pairing (FRC) model seeks to maximize the total profit, which is defined by the revenues from airfares and convenient through-flight combinations minus the operational costs of aircraft and crews as well as the opportunity cost due to spilled customer demands.

We designed a two-phase algorithm for solving Problem FRC, where in Phase I, a partially integrated Model FAAR-RLT that involves the fleet assignment and aircraft routing decisions was solved, and the resulting fleet assignment decisions were then passed to the crew pairing problem. In Phase II, we next solved the fully integrated Model FRC using a Benders decomposition-based approach, where the fleeting and routing decisions were handled within the Benders master program, and the passenger-mix and crew pairing decisions were treated within separable subproblems. Furthermore, the crew pairing subproblem was solved using column generation, where the restricted master program was solved using a deflected subgradient algorithm as proposed by Subramanian and Sherali (2008). In addition, several acceleration strategies were applied to facilitate the solution process, such as using the strategy of McDaniel and Devine (1977) to generate an initial set of Benders cuts, that of Geoffrion and Graves (1974) to solve the Benders master program as a single mixed-integer program, and the technique for generating maximal nondominated Benders cuts by appropriately perturbing the right-hand sides of subproblems as per Sherali and Lunday (2011).

In order to demonstrate the efficacy of the proposed integrated model and the solution framework, several test instances were derived from real-life data provided by a leading US legacy carrier, *United Airlines*. The computational results revealed the effectiveness of the partially integrated approach in Phase I whereby near-optimal solutions to instances involving up to 500 daily flights were readily obtained within five wall-clock hours. Furthermore, the fully integrated Model FRC solved in Phase II achieved an average improvement in total profit of 2.73% over the Phase I solution within reasonable times, which translates to about 43 million dollars per year. This improvement resulted from a better utilization of the available fleets, whereby relatively larger-sized jets were deployed more judiciously to facilitate crew pairing decisions, thus yielding correspondingly lower crew costs that compensated for the slightly increased aircraft operational costs, leading to an overall net increase in profit.

The flexibility of the proposed model permits incorporating other aircraft routing restrictions without adding much expense in terms of the model size. The crew pairing rules can also be readily modified in order to accommodate different realistic requirements. Moreover, other airline operational decision-making stages can be further integrated within the framework, such as the *demand-driven dispatch* strategy (Shebalov, 2009) that examines provisional plane swaps according to updated demand information. Furthermore, the proposed framework can be used to study other higher-level decisions, as for example, the potential benefits of incorporating certain optional flight legs, and can be tied in with revenue management strategies in order to better deal with seat inventory control, as well as to provide insights into pricing and overbooking decisions. Finally, another possible direction for future research is to enhance the model's solvability from the computational perspective by using parallel computing techniques, particularly since the Benders subproblems are solved independently from each other.

Bibliography

Abara, J. 1989. Applying Integer Linear Programming to the Fleet Assignment Problem. *Interfaces* **19**(4) 20–28.

AhmadBeygi, S., A. Cohn, M. Weir. 2009. An Integer Programming Approach to Generating Airline Crew Pairings. *Computers & Operations Research* **36**(4) 1284–1298.

Barnhart, C., N. L. Boland, L. W. Clarke, E. L. Johnson, G. L. Nemhauser, R. G. Shenoi. 1998a. Flight String Models for Aircraft Fleeting and Routing. *Transportation Science* **32**(3) 208–220.

Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, P. H. Vance. 1998b. Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research* **46**(3) 316–329.

Barnhart, C., T. S. Kniker, M. Lohatepanont. 2002. Itinerary-Based Airline Fleet Assignment. *Transportation Science* **36**(2) 199–217.

Bélanger, N., G. Desaulniers, Soumis F., J. Desrosiers. 2006. Periodic Airline Fleet Assignment with Time Windows, Spacing Constraints, and Time Dependent Revenues. *European Journal of Operational Research* **175**(3) 1754 – 1766.

Benders, J. F. 1962. Partitioning Procedures for Solving Mixed-Variables Programming Problems. *Numerische Mathematik* **4**(1) 238–252.

Bureau of Economic Analysis. 2011. Gross Domestic Product (GDP) by Industry Data. URL `http://www.bea.gov/industry/gdpbyind_data.htm`.

Bureau of Labor Statistics. 2011. Occupational Employment and Wages Statistics: Airline Pilots, Copilots, and Flight Engineers. URL `www.bls.gov/oes/current/oes532011.htm`.

Bureau of Transportation Statistics. 2011. TranStats–Multimodal Transportation Database. Research and Innovative Technology Administration. URL `http://www.transtats.bts.gov/`.

Clarke, L., E. Johnson, G. Nemhauser, Z. Zhu. 1997. The Aircraft Rotation Problem. *Annals of Operations Research* **69** 33–46.

Clarke, L. W., C. A. Hane, E. L. Johnson, G. L. Nemhauser. 1996. Maintenance and Crew Considerations in Fleet Assignment. *Transportation Science* **30**(3) 249 – 260.

Cohn, A. M., C. Barnhart. 2003. Improving Crew Scheduling by Incorporating Key Maintenance Routing Decisions. *Operations Research* **51**(3) 387–396.

Cordeau, J.-F., G. Stojković, F. Soumis, J. Desrosiers. 2001. Benders Decomposition for Simultaneous Aircraft Routing and Crew Scheduling. *Transportation Science* **35**(4) 375–388.

Daskin, M. S., N. D. Panayotopoulos. 1989. A Lagrangian Relaxation Approach to Assigning Aircraft to Routes in Hub and Spoke Networks. *Transportation Science* **23**(2) 91–99.

De Merle, O., D. Villeneuve, J. Desrosiers, P. Hansen. 1999. Stabilized Column Generation. *Discrete Mathematics* **194**(1) 229–237.

Desaulniers, G., J. Desrosiers, Y. Dumas, M. M. Solomon, F. Soumis. 1997. Aircraft Routing and Scheduling. *Management Science* **43**(6) 841–855.

Desrochers, M., F. Soumis. 1989. A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science* **23**(1) 1–13.

Desrosiers, J., Y. Dumas, M. M. Solomon, F. Soumis. 1995. Time Constrained Routing and Scheduling. *Handbooks in Operations Research and Management Science* **8** 35–139.

Dorn, W. S. 1960. A Duality Theorem for Convex Programs. *IBM Journal of Research and Development* **4**(4) 407–413.

Elf, M., Jünger M., Kaibel V. 2003. Rotation Planning for the Continental Service of a European Airline. W. Jäger, Krebs H.-J., eds., *Mathematics - Key Technology for the Future. Joint Projects between Universities and Industry*. Springer-Verlag, 675–689.

Feo, T. A., J. F. Bard. 1989. Flight Scheduling and Maintenance Base Planning. *Management Science* **35**(12) 1415–1432.

Fleury, M. 1883. Deux Problèmes de Géométrie de Situation. *Journal de Mathématiques Élémentaires* 257–261.

Geoffrion, A. M., G. W. Graves. 1974. Multicommodity Distribution System Design by Benders Decomposition. *Management Science* **20**(5) 822–844.

Gopalan, R., K. T. Talluri. 1998. The Aircraft Maintenance Routing Problem. *Operations Research* **46**(2) 260–271.

Handler, G. Y., I. Zang. 1980. A Dual Algorithm for the Constrained Shortest Path Problem. *Networks* **10**(4) 293–309.

Hane, C. A., C. Barnhart, E. L. Johnson, R. E. Marsten, G. L. Nemhauser, G. Sigismondi. 1995. The Fleet Assignment Problem: Solving a Large-Scale Integer Program. *Mathematical Programming* **70**(1) 211–232.

Haouari, M., N. Aissaoui, F. Z. Mansour. 2009. Network Flow-Based Approaches for Integrated Aircraft Fleeting and Routing. *European Journal of Operational Research* **193**(2) 591–599.

Haouari, M., S. Shao, H. D. Sherali. 2011a. A Lifted Compact Formulation for the Daily Aircraft Maintenance Routing Problem. *Transportation Science*. Doi: 10.1287/trsc.1120.0433.

Haouari, M., H. D. Sherali, F. Z. Mansour, N. Aissaoui. 2011b. Exact Approaches for Integrated Aircraft Fleeting and Routing at TunisAir. *Computational Optimization and Applications* **49**(2) 213–239.

Ioachim, I., J. Desrosiers, F. Soumis, N. Belanger. 1999. Fleet Assignment and Routing with Schedule Synchronization Constraints. *European Journal of Operational Research* **119**(1) 75–90.

Jacobs, T. L., B. C. Smith, E. L. Johnson. 2008. Incorporating Network Flow Effects into the Airline Fleet Assignment Process. *Transportation Science* **42**(4) 514 – 529.

Klabjan, D. 2005. Large-Scale Models in the Airline Industry. G. Desaulniers, J. Desrosiers, M. M. Solomon, eds., *Column Generation*. Springer, New York, NY, 163–196.

Klabjan, D., E. L. Johnson, G. L. Nemhauser, E. Gelman, S. Ramaswamy. 2002. Airline Crew Scheduling with Time Windows and Plane-Count Constraints. *Transportation Science* **36**(3) 337–348.

Kniker, T. 1998. Itinerary Based Airline Fleet Assignment. Ph.D. dissertation, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.

Lan, S., J. Clarke, C. Barnhart. 2006. Planning for Robust Airline Operations: Optimizing Aircraft Routings and Flight Departure Times to Minimize Passenger Disruptions. *Transportation Science* **40**(1) 15–28.

Liang, Z., W. A. Chaovalitwongse. 2011. A Network-Based Model for Weekly Aircraft Maintenance Routing with Integrated Fleet Assignment Problem. *Transportation Science*. Doi: 10.1287/trsc.1120.0434.

Liang, Z., W. A. Chaovalitwongse, H. C. Huang, E. L. Johnson. 2011. On a New Rotation Tour Network Model for Aircraft Maintenance Routing Problem. *Transportation Science* **45**(1) 109–120.

Lohatepanont, M., C. Barnhart. 2004. Airline Schedule Planning: Integrated Models and Algorithms for Schedule Design and Fleet Assignment. *Transportation Science* **38**(1) 19–32.

Magnanti, T. L., R. T. Wong. 1981. Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. *Operations Research* **29**(3) 464–484.

Mansour, F. Z., M. Haouari, H. D. Sherali, N. Aissaoui. 2010. Flexible Aircraft Fleeting and Routing at TunisAir. *Journal of the Operational Research Society* **62**(2) 368–380.

McDaniel, D., M. Devine. 1977. A Modified Benders' Partitioning Algorithm for Mixed Integer Programming. *Management Science* **24**(3) 312–319.

Mercier, A. 2008. A Theoretical Comparison of Feasibility Cuts for the Integrated Aircraft-Routing and Crew-Pairing Problem. *Transportation Science* **42**(1) 87–104.

Mercier, A., J.-F. Cordeau, F. Soumis. 2005. A Computational Study of Benders Decomposition for the Integrated Aircraft Routing and Crew Scheduling Problem. *Computers & Operations Research* **32**(6) 1451–1476.

Mercier, A., F. Soumis. 2007. An Integrated Aircraft Routing, Crew Scheduling and Flight Retiming Model. *Computers & Operations Research* **34**(8) 2251–2265.

Papadakos, N. 2009. Integrated Airline Scheduling. *Computers & Operations Research* **36**(1) 176–195.

Rexing, B., C. Barnhart, T. Kniker, A. Jarrah, N. Krishnamurthy. 2000. Airline Fleet Assignment with Time Windows. *Transportation Science* **34**(1) 1–20.

Rushmeier, R. A., S. A. Kontogiorgis. 1997. Advances in the Optimization of Airline Fleet Assignment. *Transportation Science* **31**(2) 159–169.

Ryan, D. M., B. A. Foster. 1981. An Integer Programming Approach to Scheduling. A. Wren, ed., *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling*. North-Holland, 269–280.

Sandhu, R., D. Klabjan. 2007. Integrated Airline Fleeting and Crew-Pairing Decisions. *Operations Research* **55**(3) 439–456.

Sarac, A., R. Batta, C. Rump. 2006. A Branch-and-Price Approach for Operational Aircraft Maintenance Routing. *European Journal of Operational Research* **175**(3) 1850–1869.

Savelsbergh, M., M. Sol. 1998. DRIVE: Dynamic Routing of Independent Vehicles. *Operations Research* **46**(4) 474–490.

Shebalov, S. 2009. Practical Overview of Demand-Driven Dispatch. *Journal of Revenue & Pricing Management* **8**(2) 166–173.

Sherali, H. D., W. P. Adams. 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.

Sherali, H. D., W. P. Adams. 1994. A Hierarchy of Relaxations and Convex Hull Characterizations for Mixed-Integer Zero-One Programming Problems. *Discrete Applied Mathematics* **52**(1) 83 – 106.

Sherali, H. D., W. P. Adams, P. J. Driscoll. 1998. Exploiting Special Structures in Constructing a Hierarchy of Relaxations for 0-1 Mixed Integer Problems. *Operations Research* **46**(3) 396 – 405.

Sherali, H. D., K.-H. Bae, M. Haouari. 2010. Integrated Airline Schedule Design and Fleet Assignment: Polyhedral Analysis and Benders' Decomposition Approach. *INFORMS Journal on Computing* **22**(4) 500–513.

Sherali, H. D., K.-H. Bae, M. Haouari. 2011. A Benders Decomposition Approach for an Integrated Airline Schedule Design and Fleet Assignment Problem with Flight Retiming, Schedule Balance, and Demand Recapture. *Annals of Operations Research*. Doi: 10.1007/s10479-011-0906-3.

Sherali, H. D., E. Bish, X. Zhu. 2006. Airline Fleet Assignment Concepts, Models, and Algorithms. *European Journal of Operational Research* **172**(1) 1–30.

Sherali, H. D., B. J. Lunday. 2011. On Generating Maximal Nondominated Benders Cuts. *Annals of Operations Research*. Doi: 10.1007/s10479-011-0883-6.

Sriram, C., A. Haghani. 2003. An Optimization Model for Aircraft Maintenance Scheduling and Re-Assignment. *Transportation Research, Part A* **37**(1) 29–48.

Subramanian, S., H. D. Sherali. 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.

Talluri, K. T. 1998. The Four-Day Aircraft Maintenance Routing Problem. *Transportation Science* **32**(1) 43–53.

Weide, O., D. Ryan, M. Ehrgott. 2010. An Iterative Approach to Robust and Integrated Aircraft Routing and Crew Scheduling. *Computers & Operations Research* **37**(5) 833–844.

Yan, S. 2002. A Passenger Demand Model for Airline Flight Scheduling and Fleet Routing. *Computers & Operations Research* **29**(11) 1559–1581.

Yu, G. 1998. *Operations Research in the Airline Industry*. Springer, New York, NY.