INTEGRATED AIRLINE PLANNING MODELS

by

Anne Elizabeth Catlin Johnson

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2 0 0 5

The University of Arizona ®
Graduate College

As members of the Final Examination Committee, we certify that we have read the dissertation prepared by Anne Elizabeth Catlin Johnson entitled Integrated Airline Planning Models and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

____ 8 April 2005
Julia L. Higle

____ 8 April 2005
Suvrajeet Sen

____ 8 April 2005
J. Cole Smith

____ 8 April 2005
Arthur L. Wright

____ 8 April 2005
Moshe Shaked

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

____ 8 April 2005
Dissertation Director: Julia L. Higle

STATEMENT BY AUTHOR

SIGNED: Anne E. C. Johnson

ACKNOWLEDGEMENTS

First and foremost, I am indebted to my advisor, Dr. Julie Higle, for her guidance, support, and patience throughout the entire process. I could not have asked for a better advisor. She helped me maintain a long-term perspective for launching an academic career as well as the short-term goal of completing my degree, and was extremely supportive of my self-imposed timeline.

My other committee members, Drs. Suvrajeet Sen and Cole Smith, were responsible for greatly expanding my understanding of optimization through their courses. They provided valuable feedback that contributed to my success.

Drs. T. Glenn Bailey, Milind Sohoni, Wassim Chaar, and others at Delta Technology were gracious enough to invite us out for meetings in the summer of 2003. Though a formal research partnership never came to fruition, they provided a wealth of questions, ideas and information, much of which is included in this dissertation.

A significant portion of my work was built on code written by fellow SIE graduate student Lei Zhao for a large-scale optimization course in the fall of 2003. My C programming knowledge and skill improved considerably after picking through his code for months on end, and I am very grateful that he let me use it.

A few special groups of people in the SIE department deserve recognition. I must thank my officemates for helping in the morale department, especially fellow statistics minor Jenn Barzeele. Also, the support staff was very helpful, especially Linda Cramer for ensuring I jumped through the appropriate wickets, and Bill Ganoe for assistance with computer issues.

Finally, I am eternally grateful to my husband, Lt Col Rick Johnson, who made considerable sacrifices in job satisfaction so that I could return to school. He tolerated my odd hours in front of the screen, and encouraged me when I got completely fed up. He even took a two-year assignment “flying a desk” instead of an airplane, where he belongs, so that we could stay in Tucson long enough for me to finish.

This research was funded by Grant No. DMS-0400085 from the National Science Foundation.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Technological and industrial advances have resulted in the growth of large enterprises. Optimization models have been developed to increase the efficiency of parts of these systems, but models that optimize entire enterprises are frequently immense and very complex to solve. Sequential solution techniques have resulted, which lead to useful, but not globally optimal, solutions. For example, airlines develop flight schedules based on strategic business objectives, and sequentially plan operational processes to execute the schedule. Proven models that exist for the operational subproblems are solved sequentially, begin with a flight schedule, and allow limited feedback in the planning process. Since small changes to the individual parts have produced millions of dollars in improvement, an overall optimal solution could yield a significant increase in the airline's profit.

We consider a modelling paradigm that moves toward integrated methods for the airline schedule planning phase using surrogate representations of the operational problems. In this context, surrogate models are relatively easy to solve, yet sufficiently representative of the operational problem to reflect its impact on schedule choices. To illustrate, we develop surrogate models of maintenance scheduling, crew scheduling, and revenue generation. We solve the master schedule problem with each surrogate model using well-known decomposition techniques, and then combine the surrogates into a single model that is readily decomposed into multiple subproblems and solved.

The model developments include additional considerations in constructing surrogate models. For example, to demonstrate validation of a surrogate's utility, we compared the feasibility indications from the maintenance subproblem surrogate to

those from a larger, exact model of maintenance feasibility. The crew scheduling surrogate model development incorporates disruptions in the master schedule, driving the schedule to account for both crew costs and the impact of random disruptions. Finally, in the revenue management subproblem, we consider random demand that impacts a schedule's profitability.

While surrogate solutions are inherently of little utility operationally, the results are useful for shaping the master schedule towards a global optimum. The paradigm allows for consideration of the subproblems in initial planning, so that solutions obtained from the full models are based on a schedule that may lead to a better overall bottom line.

CHAPTER 1

Introduction

Advances in technology and industry have resulted in the growth of large enterprises, particularly in the last fifty years. Enterprise systems exist in manufacturing, transportation, telecommunications, and other fields. Operations research evolved largely to increase the efficiency and improve control of such systems, as the manual methods that work for small endeavors are cumbersome, if not impossible, to use on a grand scale.

When an enterprise becomes large, the models used to optimize the entire operation become intractable. Even with immense computational power, solution techniques are often not advanced enough to solve extremely large, multifaceted enterprise problems. Therefore, analysts often optimize smaller pieces of the system independently. If one part of the system depends on the outcome of another and the pieces are examined sequentially, solutions that are good and useful, but are probably not globally optimal, tend to result.

Major airlines exemplify large enterprises that use sequential techniques in planning. Airlines develop flight schedules, crew schedules, and other cost drivers sequentially, and often adjust the solutions from previous parts of the system without feedback to those subsystems. Although many parts of the airline enterprise have benefitted from considerable research in solution methods, the sequential solution of the parts still results in an overall loss of potential profit. Since small changes to the individual parts have yielded millions of dollars in improvement, an overall optimal solution would probably lead to a significant increase in the airline's bottom line. The operations research section of a major airline expressed specific interested in integrated scheduling during a 2003 discussion.

Airlines are thus quite interested in integrated planning techniques. While some partially-integrated techniques have recently been developed, methods to find an overall optimal schedule remain elusive. This dissertation presents an investigation of novel models and techniques that treat the airline planning problem in an integrated manner. While some of the models are inexact approximations of the real system, we believe that they can be useful in steering an airline's planning efforts, and can find schedules that enable profit maximization.

We will begin with an examination of relevant operations research methods and literature, as applied to airline planning. Next, we develop the master schedule problem, followed by an exploration of a surrogate model for maintenance scheduling. We follow these discussions with examinations of surrogate models for crew scheduling and revenue generation, and then present an integrated model that considers all of the parts to find a profit-maximizing schedule.

CHAPTER 2

Airline Operations Research

2.1 Introduction

While the bulk of optimization research as applied to airlines has appeared in the past fifteen to twenty years, the foundations for network modelling date back a few decades further. Ford and Fulkerson published *Flows in Networks* in 1962, a definitive work that summarizes progress in network modelling and transportation problems to that point. Since then, computational power limitations have driven researchers to seek efficient methods for finding optimal and near-optimal solutions to network problems, as well as general integer and combinatorial problems. This technology has been applied to telecommunications networks, urban transportation, manufacturing, banking and finance, and numerous other areas as well as airline planning and operations.

This review will explore dynamic and multicommodity network models relevant to airline optimization, both deterministic and stochastic. Since most interesting network problems are large and consequently difficult to solve with traditional methods such as the standard simplex method, we include the computational methods developed for solving these network problems, as well as algorithms for handling uncertainty in a problem, in Appendix A. A survey of airline applications will follow.

2.2 Airline Operations Research

The large and complex nature of airline networks, as well as the potentially huge impact of small changes in operations on an airline's bottom line, make airlines a fertile area for optimization research. Airline operations consist of a large schedule with several challenging subproblems. Initial route selection depends on passenger demand forecasts, which can change frequently as departure dates approach. An aircraft type, or *fleet*, must be assigned to each scheduled flight, and the flight schedule is constrained by aircraft availability. Each flight must have a crew trained to fly the assigned aircraft, and each crew must start and end their trip at their *base*, or home airport. Crews must also follow FAA rules regarding maximum hours of flying per day, and must maintain training and currency standards. Aircraft require periodic maintenance, some of which can only be performed at designated airports. Since the overall objective of airline operations is to maximize profit, all of these subproblems must aim at minimizing costs and maximizing revenue.

Most research to date has focused on the subproblems independently or sequentially. For example, fleeting depends on route selection, and crew scheduling is based on fleet assignment and route schedule. A discussion of each subproblem and related research efforts follows. Note that sequential solutions of subproblems generally move away from the global optimum with each step, as an optimal solution for one subproblem rarely coincides with another subproblem's optimal solution.

2.2.1 Revenue management

The overall objective of airline optimization is usually to maximize profit, though some airlines still aim to maximize revenue. Airlines generate revenue by selling tickets, but nearly every airline sells different seats on the same flight at different prices. The prices correspond to different levels of service, or *fare classes*. The fare class of a ticket reflects first or coach class, whether the passenger may exchange or return the ticket without paying a fee, how far in advance the ticket was sold, and other factors. Different passengers are interested in different prices and levels of service, and some are willing to accept a higher or lower fare class than originally requested. Seats are treated as perishable products by the airlines, since seats can only be sold before the flight's departure. Some customers buy tickets earlier than others, and while fare class and purchase time are correlated - vacationers usually buy nonchangeable coach class tickets well in advance to pay lower fares, whereas business passengers generally buy more expensive flexible and first-class tickets only a few days before the flight - the timing of ticket purchases exhibits stochastic behavior.

This multidimensional randomness has led to considerable research in the area of *revenue management* for airlines, hotels, car rentals, and other industries. McGill and Van Ryzin (1999) and Pak and Piersma (2002) present surveys of airline revenue management research. Various efforts model the demand for each fare class as stochastic or deterministic, using continuous or discrete time. *Dynamic* revenue management methods reoptimize the allocation of seats on a flight to fare classes after every booking, while *static* methods allocate seats once or at a few selected *decision points* in time. Revenue management discussions often use the following terms:

Protection level The number of seats allocated to each fare class that may not be reassigned to a lower fare class, but are instead reserved for customers paying the higher fare.

Nested classes A policy system that sets protection levels for each class and all classes above it. Therefore, a seat sales limit exists for the lowest fare class, the two lowest classes together, the three lowest classes together, and so forth. This allows an airline to sell lower fare seats to higher fare customers if the higher class sells out first, but protects higher-fare seats from being sold to lower fare customers. Some airlines use a variation of nested classes in which revenue managers calculate the expected revenue of a seat in the next higher class by multiplying the seat price and the probability of selling that seat in the assigned class. They may then violate the protection levels if the expected revenue is higher when the seat is sold to the low-fare customer than when the seat remains protected and may therefore fly empty.

Batch and group booking, overbooking, cancellations, and no-shows

Batch booking refers to simultaneous purchases of two or more tickets. Group bookings are batch bookings at a discounted rate, such as tour group bookings. Overbookings are airline policies of selling extra seats on a flight, expecting that some passengers will cancel their trips (cancellations) or not show up for the flight (no-shows). These four anomalies are assumed out of many revenue management models, as myriad scenarios exist for each and a relatively small fraction of passengers fall into the categories. For example, a no-show may have a refundable ticket or may have switched to a different flight immediately before departure. Accounting for the loss of revenue or transfer of revenue between flights is very complex. Subramanian et al. (1999) present a revenue management model that includes overbooking, cancellations, and no-shows, and they find that a 9% increase in revenue is possible if these factors are included.

Demand recapturing and spill cost A passenger attempting to purchase a ticket in a sold-out fare class may buy a ticket in a higher or lower class, if available, or may take a different flight. Spill cost refers to the lost revenue of customers who purchase no ticket, or spill to another airline, when their desired fare class and flight are unavailable. Rexing et al. (2000) and Barnhart et al. (2002) discuss spill costs in the design of their capacity assignment models.

Single-leg and network demand A *flight leg* is a single nonstop flight between two airports. In reality, most passengers travel on a multi-leg itinerary and are interested in getting from an origin to a destination, or *O-D pair*. Customers will often accept a flight connection or a *through flight*, which is a flight with a stop where the passenger stays on the same airplane[1]. The intermediate

[1] Through flights can be more profitable since some passengers, particularly higher-paying business travellers, are willing to pay a premium to avoid changing planes. Airlines generally schedule a through flight between major business travel destinations when a nonstop flight is infeasible or impractical at a certain time of day.

city is generally unimportant to the customer. Demand data based on origin-destination pairs are called *network demand* data, or simply O-D demand data. Since network demand can be computationally expensive to process, many models use single-leg demand, which assumes that passengers purchase tickets for point-to-point travel with no stops or connections. This simplified approach ignores the fact that a ticket from city A to city B via city C is usually less expensive than two separate tickets from A to C and C to B. Single-leg demand can also overestimate demand to and from major connection airports known as *hubs*. *Itinerary demand*, or demand based on an O-D pair with one or more specific intermediate stops, is also used at times, but can also be inaccurate since most travellers are not concerned about the intermediate stop as long as they get from an origin to a destination.

Revenue maximization models are important in assigning aircraft types to a flight schedule to optimize use of available capacity. After *fleeting* the schedule, or assigning aircraft types, revenue management continues to change fare class allocations as ticket sales progress. Some models allow re-fleeting if swapping a large, undersold airplane for a small, overbooked airplane will increase revenue, but such changes generally occur well before flight departure and will result in modifications to the crew and maintenance schedules as well. Good demand forecast data and revenue management models can reduce the need for reassigning capacity later in the planning process.

2.2.2 Demand modelling

Until recently, some major airlines have modelled single-leg demand without accounting for network effects. This method may lead to over- or underestimation of demand on some legs. For example, if a passenger wishes to travel from Tucson (TUS) to Baltimore (BWI), she may choose between connections in Atlanta (ATL) or Cincinnati (CVG). If she chooses an itinerary of TUS-ATL, ATL-BWI, the demand leg model accounts for her purchase as demand on the TUS-ATL and

ATL-BWI legs. She may have been equally likely to choose the Cincinnati routing, and a slightly better demand model would account for this possibility. The most accurate model would count the passenger's demand for travel from TUS to BWI instead of the separate legs, as the passenger really is not interested in any of the legs through the connection cities, provided she gets from the origin to the destination.

Demand is dependent on time as well as O-D pair. Vacationers tend to purchase tickets well in advance of flights in order to pay the lowest fares. Business travelers often buy expensive flexible or first-class tickets within a week or less of the flight departure date. Therefore, demand for "cheap seats" is highest early in the ticket sales period, and demand for expensive tickets escalates as the flight date nears. Further, most airlines sell the least expensive tickets only up to 21 or 14 days before a flight. External events may impact this behavior as well; for example, many business travelers cancelled trips after the September 11, 2001 terrorist attacks and the subsequent souring of the economy, so full fare ticket sales close to flight dates were much lower than expected. Cancellations and overbooking policies complicate matters further, requiring the airlines to predict more random factors and to react with rebookings on the day of travel.

Airline bookings can be modelled as discrete demands at discrete time points, which allows multistage decision making to account for time dependency to some extent. For example, Higle and Sen (1997) describe a two-stage model for airline revenue with random demand levels arising at selected times before a flight. In the first stage, they make an initial capacity allocation decision by flight leg and class. In the second stage, they assign leg capacity to O-D itineraries, allowing seats on each leg to be reassigned from one itinerary to another to improve profitability. They acknowledge that the model cannot reassign capacity between classes, and suggest modifications to allow seat reassignment to a different class.

Though airlines would ideally produce exactly the correct number of seats to satisfy passenger demand, they are generally constrained to selling tickets according to the seat capacity of their airplanes. Airlines may add or remove seats from planes as part of a long-term plan, but the number of seats in each cabin on a plane can be considered fixed for schedule planning purposes. Therefore, assigning airplane types to flight legs to best match demand is another important step in developing a maximally profitable flight schedule.

2.2.3 Capacity and fleet assignment

An airline begins schedule planning by scheduling flight legs from one *station*, or airport, to another. This first plan is based roughly on historical schedules and demand and contains many legs which will not appear in the final schedule. Given this initial flight schedule, the airline assigns capacity (aircraft) to each flight leg to meet forecasted passenger demand. This process is called *fleet assignment*, and involves maximizing the *fleeting contribution* given aircraft availability and flow conservation constraints. The fleeting contribution is defined as the maximum attainable revenue for a flight leg, less the *assignment cost*. The assignment cost is the flight operating cost, passenger carrying cost, and *spill cost*, or opportunity cost for passengers not accommodated by the flight, according to Barnhart et al. (2002). Spilled passengers can be *recaptured* by other flights on the same airline, perhaps at a different time or even on a different day. Some flight legs may assume a capacity of zero if aircraft are not available, or if the cost of capacitating a leg outweighs the revenue. This point highlights the need to examine the impact of flight connections on overall profit, as airlines may choose to fly an unprofitable flight leg in order to reap more revenue on a connecting leg. Traditional fleet assignment models generally ignore this property of airline networks.

To capture the effect of connecting flights on profit and to accommodate scheduled maintenance requirements, Barnhart et al. (1998) propose *flight string models*. They define a flight string as a "sequence of connected flights that begins and ends at maintenance stations, satisfies flow balance...and is maintenance feasible." Since any realistic flight schedule contains an intractable number of feasible flight strings, they employ a branch-and-price algorithm to limit the number of variables and thus the size of the matrix required by the solution method.

Most domestic airlines currently use a daily fleet assignment model (FAM) proposed by Hane et al. (1995) and discussed by Barnhart et al. (2002). This large multi-commodity flow model includes side constraints to enforce required through flights. The FAM is as follows, where F is the set of fleets, K is the set of flights, N_k is the number of planes in fleet k, M^k is the set of nodes in fleet k's network, and H^k is the set of ground arcs in fleet k's network. Network-defining data variable $b1_{nik}$ is 1 if flight i begins at node n in fleet k's network, is -1 if flight i ends at node n in fleet k's network, and 0 otherwise. Similarly, $b2_{nhk}$ is 1 if ground arc h begins at node n in fleet k's network and -1 if it ends there, and is 0 otherwise. Finally, $d1_{ik}$ is 1 if flight i crosses the *count time* while airborne in fleet k's network, and is 0 otherwise, and $d2_{hk}$ is the equivalent for ground arcs. (The count time is the renewal time of the schedule, typically 3:30am for a daily schedule.) The decision variable $x_{ik} = 1$ if fleet k is assigned to flight i, and zero otherwise, while w_{hk} represents the number of planes in fleet k held on ground arc h. The cost coefficient c_{ik} is the operating cost of fleet k for flight i plus the opportunity cost of spilling passengers if this fleet assignment occurs.

$$\min \sum_{k \in K} \sum_{i \in F} c_{ik} x_{ik} \tag{2.1a}$$

$$\text{s.t.} \quad \sum_{k \in K} x_{ik} = 1 \qquad \forall\, i \in F \tag{2.1b}$$

$$\sum_{i \in F} b1_{nik} x_{ik} + \sum_{h \in H^k} b2_{nhk} w_{hk} = 0 \qquad \forall\, n \in M^k, k \in K \tag{2.1c}$$

$$\sum_{i \in F} d1_{ik} x_{ik} + \sum_{h \in H^k} d2_{hk} w_{hk} \leq N_k \qquad \forall\, k \in K \tag{2.1d}$$

$$x_{ik} \in \{0, 1\} \qquad \forall\, i \in F, k \in K \tag{2.1e}$$

$$w_{hk} \geq 0, w_{hk} \in Z^+ \qquad \forall\, h \in H^k, k \in K \tag{2.1f}$$

To reduce the problem size, the model aggregates groups of flights which arrive and depart near the same time into nodes called *islands*. The LP solution algorithm then prescribes a sequence of cost perturbation, dual steepest-edge simplex, perturbation removal, reoptimization, disaggregation, fixing of flight variables with values greater than 0.99 to 1, reaggregation, and resolving with dual steepest-edge simplex and intelligent branch-and-bound. An interior point method follows a similar sequence but includes a primal-dual predictor-corrector instead of the first dual simplex step. In most cases, the LP method was faster than the interior-point method, but both solved realistic eleven-fleet, 2,500-flight daily problems in under an hour. The IP-LP optimality gap was less than 1 percent in all cases, using real airline data.

In this procedure, the objective function requires deterministic treatment of volatile data; for example, the model estimates potential revenue for a flight by multiplying the assigned fleet's seat capacity by the average fare and the minimum predicted demand. Both the demand forecasts and the average fare can vary significantly and will fluctuate between planning time and flight time, so this fleeting algorithm provides a useful solution but rarely an optimal one. Also, as noted by Barnhart et al. (2002), the procedure fails to account for spill cost and recapture. The authors propose an itinerary-based fleet assignment model (IFAM) that accounts for spill costs and recapture using expected value and the *Quantitative Share Index*, a measure of market share used to estimate the probability of a customer purchasing one airline's itinerary instead of all others in a market. The solution algorithm iteratively adds columns and violated constraints to a restricted master problem (RMP), generates cuts to partition spill costs into positive and zero sets, and resolves the RMP until no more rows or columns are added. A heuristic branch-and-bound step completes the algorithm. While added accuracy from considering spill and recapture significantly improves the accuracy of the FAM, the data is still deterministic and based on expected value. Barnhart et al. (2002) justify the use of averages with the requirement for operational stability, since a schedule that changes to accommodate daily demand fluctuations would be infeasible or extremely difficult to execute.

Rexing et al. (2000) also acknowledge spill costs and stochasticity in demand data, and suggest approaches for estimating demand accordingly. They note that flight times usually change at least slightly in the months between initial schedule development and takeoff, and suggest that considerable cost savings may arise when flight departure times are adjusted within small time windows. The time window model contains an arrival node for each flight, which occurs after the arrival time and the required ground time, as well as a node for all possible departure times for each flight within the time window. Therefore, the model includes many more nodes than the standard FAM. An iterative solution technique resembles that of Hane et al. (1995), but after each run, the flight network is recreated with one reduced-duration arc for each flight-fleet pair, and the solver attempts to restore feasibility and then reoptimize. This approach led to the refleeting of 10 to 20 percent of flights, and lowered the total number of expensive aircraft required as well as the fleet assignment cost in several experiments with real data.

While FAM, IFAM and the time window approaches include heuristic techniques as part of the branch-and-bound step of each algorithm, Ahuja et al. (1993) propose a method based primarily on heuristics. The *through assignment model* (TAM) builds through flights into a FAM solution, and then considers opportunities for pairwise fleet exchanges on selected legs. In a network with n fleets, $n(n-1)$ pairs of fleet types may be swapped to search for improvement. The neighborhood search procedure iteratively swaps pairs of fleets and retains those which yield reduced cost, until all combinations have been examined and no more profitable swaps can be found. This procedure terminated in 6 seconds for a 13-fleet problem and was estimated to yield a $25 million cost reduction for United Airlines. While the method is overly simplistic and admittedly built on questionable assumptions, it represents a good initial application of heuristics to one aspect of airline optimization.

The fleeting approaches discussed so far are all oriented toward the planning phase of scheduling, while considerable analysis and tuning of the schedule occurs following initial plan development. Fleeting assignments may change to accommodate demand forecast fluctuations, aircraft availability, and other operational or business-related concerns. Jarrah (2000) present several approaches for quick re-fleeting, and accounts for through flights, maintenance opportunities, crew staffing levels, and even noise restrictions[2] with side constraints. They group aircraft into fleet types, based on size, and aggregate islands, and solve the LP relaxation with dual steepest-edge simplex as described previously. However, they employ a set of modules to allow schedule changes which may improve profitability. The modules include:

Popping Finds the least expensive way to remove a specified number of aircraft from the schedule

Change-of-Gauge Permits the replacement of a selected number of aircraft in one size class, or gauge, to be replaced with the same number from another class, thereby changing capacity

Swapping Allows a limited number of fleet-leg assignment swaps within a size class

Utilization Shifts *block time*, or total time from departure to arrival, between two classes when the schedule for one class exceeds crew availability, in order to increase crew utilization and/or reduce the number of crews

The algorithm iteratively estimates lower bounds, defines special ordered sets for branch-and-bound, and performs different types of branch-and-bound.

Airline fleeting models have clearly improved considerably in recent years. However, their approaches to cost estimation are still rough, especially considering ran-

[2]Some airports restrict the amount of older, louder aircraft traffic allowed per day, as well as their hours of operation.

dom demand based on a whole itinerary as opposed to a individual legs. Also, none of the research described accounts for the fact that different fleets fly at different speeds, so that a fleeting swap may require a schedule change. While flight time difference may not be significant between wide-bodied and newer 737-800 aircraft, a regional jet or older airplane can require an extra half hour or more to complete a medium-range flight (for example, from Dallas to Tucson), which could change the set of feasible schedules. Further, the relative cost of operating more newer, smaller planes, that fly more slowly, and fewer of the larger, faster, more operationally expensive older planes has not been thoroughly investigated in the operations research literature. Such issues present an interesting problem in long-term cost analysis for fleet planning. The cost of changing major capital assets (airplanes) also includes the hiring or retraining of crews and maintenance considerations, which can be very expensive.

2.2.4 Crew scheduling

Crew scheduling has arguably received more attention than any other aspect of airline optimization, because of the extremely high cost of pilots and the complexity of the problem. FAA rules, collective bargaining agreements between airlines and pilot unions, and regular pilot training requirements impose numerous constraints on crew schedules. An explanation of crew scheduling terminology precedes a survey of models and solution techniques for this problem. Many of these terms apply to North American airlines, and may be different or not applicable elsewhere. For example, North American airlines normally aim to minimize overall crew cost, which is a function of numerous factors described below. Many European airlines pay their crews a flat monthly salary, which completely changes the crew scheduling objective to maximizing crew utilization, and thus minimizing the number of crews needed (Andersson et al., 1998; Arabeyre et al., 1969).

Technical crew Cockpit flight crew, rated as captains and first officers (or pilots and copilots). Some aircraft still require second officers as well. All technical crew members are FAA-certified pilots, and each flight requires a full crew.

Cabin crew Flight attendants. The number of flight attendants usually depends on the size of the aircraft, but the cabin crew size may vary as long as the flight meets FAA flight attendant-to-passenger ratio requirements.

Base Each crewmember operates from a base, or home airport. Each major airline operates only a few pilot bases but numerous flight attendant bases.

Block time "Blocks" refer to the wooden blocks, or chocks, placed under the wheels when the plane is parked. Block time includes the time from the removal of chocks and departure of a flight to arrival at the gate at the next destination, when chocks are again placed under the wheels.

Sit time Time between flights when a crew waits for the next departure. Often counts against the crew's maximum monthly work time.

Duty period A day of work, beginning when the pilot arrives for the first briefing of the day and ending with the last debrief. Includes all block time and sit time in between these briefings. Some duty periods may continue into a second day if the crew flies late into the night or overnight. Limited to twelve hours per day.

Flight time Synonymous with block time. Limited by the FAA to a total of eight hours per duty period for technical crews. Flights longer than eight hours require two technical crews.

Pairing A sequence of duty periods, usually beginning and ending at a crew base. Commonly called a "trip" by pilots. Collective bargaining agreements limit the maximum number of days per pairing.

Time away from base Self-explanatory, and often limited by collective bargaining agreements.

Desiderata Pilot requests, such as leave or specific trips, which impact a pilot's availabilty for flight assignments.

Appointments Not an official industry term; will be used for brevity to refer to all duty-related appointments, such as medical checks, flight examinations, and training in the discussions which follow.

Rostering Linking pairings into monthly work schedules for each crew member. Some airlines personalize work schedules by incorporating desiderata and appointments for each person, while others solve the rostering problem and assign the most appropriate work schedule to each crew member, swapping duties manually if needed to accommodate desiderata and appointments. Collective bargaining agreements limit the maximum number of duty days, work hours, and flying hours in a monthly work schedule.

Bidline A monthly work schedule on which crews "bid." Crews are assigned to bidlines in order of seniority and preference. Used by some airlines instead of personalized rostering.

Line crew member Crew members regularly assigned to a flight schedule.

Reserve crew member Crew member assigned to a base and available to fly on short notice, but not regularly assigned to a flight schedule. Brand-new employees usually start as reserve crews because of the unpredictability of reserve flying duty.

Deadhead A crewmember flying as a passenger while on duty, sometimes on another fleet or another airline. Airlines use deadhead legs to preposition crews for a pairing, reposition crews to recover a disrupted schedule, or to fly crewmembers home at the end of a pairing if needed. Deadheading is expensive and usually avoided in short- and medium-haul flight networks, but may be cost effective at times in long-haul networks when the deadhead leg is cheaper than the cost of accommodating crew away from their base. Barnhart et al. (1995) present methods for identifying cost-effective deadhead legs for the long-haul scheduling problem.

Minimum pay guarantee Crews are guaranteed a minimum amount of pay for pairings so that crews assigned to short duties are not penalized financially. Also, labor contracts guarantee pilot pay for a minimum number of flying hours each month, whether or not the pilot actually flies the minimum number of hours. This provision protects reserve pilots from financial hardship, but also highlights the importance of high crew utilization.

Pay-and-credit Total pay for flight time, sit time, and deadheading for a duty period. The actual cost of a duty period is the maximum of the minimum pay guarantee and pay-and-credit. Some airlines measure crew schedule efficiency by the percentage of pay-and-credit due to costs other than flight time.

Disruption A departure from the planned crew schedule. May be caused when crews arrive late from a previous flight, or no-shows due to illness or emergency, for example. Also, if a crew has experienced delays during a duty period, they may not be available for the last planned flight of the day because of flight time or duty period hour limits, although these rules may be violated in some circumstances.

Recovery Actions taken to return to a schedule following a disruption. May involve using reserve pilots, deadheading, or other actions chosen to minimize further disruption to the schedule.

Nearly all pilots are qualified to fly only in one fleet or fleet family (similar aircraft, such as B757 and B767) at a time, so each fleet's technical crew is scheduled separately. Further, first officers cannot fill captain crew positions, while the reverse is possible, but expensive. Most optimization efforts focus on technical crew scheduling. Flight attendants are less expensive, have fewer duty hour restrictions and more bases, can fly on most types of aircraft, and do not always require a full crew complement on each flight. Therefore, their schedules are less difficult to construct and recover, although the immense number of feasible pairings makes algorithmic efficiency crucial if an airline chooses to optimize cabin crew schedules similarly to technical crew schedules.

Crew scheduling problem fundamentals. The crew scheduling problem aims to construct minimum-cost crew pairings or crew rosters, depending on the model used by the airline. Crew schedules are built to staff the planned flight schedule, so that each flight is assigned a technical crew and a cabin crew. Most methods use a duty matrix in which each row corresponds to a flight leg and each column represents a pairing, so that a nonzero element indicates that the column's pairing covers the row's flight leg. A similar matrix exists for rostering, such that rows denote pairings and columns denote work schedules. In the personalized rostering case, zeros may be used to represent pilot nonavailability. Because of the immense number of feasible pairings and rosters in a major airline's schedule, matrix size reduction is essential for computational tractability. The standard crew pairing model appears as follows, where a_{ij} is 1 if flight i (of m total flights) is covered by crew pairing j (of n total crew pairings), and c_j is the cost of pairing j. The decision variable x_j is 1 if a crew is assigned to pairing j and 0 otherwise.

$$\min_j c_j x_j \tag{2.2a}$$

$$\text{s.t.} \quad \sum_j a_{ij} x_j = 1 \qquad \forall\, i = 1, \ldots, m \tag{2.2b}$$

$$x_j \in \{0, 1\} \quad \forall j = 1, \ldots, n \tag{2.2c}$$

Starting in the early 1970's, airlines reduced these pairing matrices using only preprocessing rules based on policy and other factors (Arabeyre et al., 1969). Some airlines combined flight legs to define basic planning units longer than a single flight leg[3]. Many airlines used branch-and-bound procedures to solve the reduced problem,

[3]British European Airways paid its pilots a flat monthly salary, so fewer crews meant lower costs, and longer trips improved crew utilization and reduced the number of crews. By removing

while others used various cutting plane algorithms or heuristics. More recently, column-generation, set-covering, and set-partitioning algorithms have been applied to the crew scheduling problem.

Set-covering and set-partitioning for pairing design. Early efforts to apply set-partitioning to airline crew scheduling included Rubin's decomposition algorithm. Rubin (1973) suggested choosing the most expensive two or three columns from an initial feasible solution of pairings and enumerating potential flight leg swaps within this subset, searching for cost improvements. He found that the set-covering procedure for two or three columns was rapid, while generating columns required more time. Rubin also suggested assigning a high cost to deadhead legs, making them feasible but undesirable. More recently, Anbil et al. (1992) proposed a more elegant decomposition by optimizing a small subproblem of all possible pairings, pricing out the rest of the columns, and then creating a new subproblem with the basis of the previous subproblem and the most expensive columns in the rest of the matrix. Using a subset of 5000 columns from a matrix with 5.5 million columns and about 800 rows, they were able to substantially reduce crew costs for American Airlines after 25 iterations.

United Airlines implemented a similar model using a pairing generator and an optimizer, as described by Graves et al. (1993). They present an elastic set partitioning integer program formulation in which the objective function penalizes deviations from one-to-one set coverage and manpower feasibility, in addition to minimizing cost. The pairing generator uses a series of embedded rules and user-selected parameters, such as minimal connection times, to find disjoint (nonoverlapping) pairings that cover all flight legs. The optimizer prices and selects columns for subproblems, and solves them with an algorithm of cutting planes, block echelon enumeration, and elastic programming. Unlike the Anbil et al. algorithm, in which subproblem

---

any shorter trips which were covered by longer trips from the pairing matrix, BEA reduced the number of columns from 25,000 to 3,000.

solutions replaced original pairings, the Graves et al. algorithm adds improved pairings found in the subproblems to the matrix for consideration when the problem is resolved. They assert that this reduces subproblem solution times and increases the chance of finding good pairing combinations.

Andersson et al. (1998) present another modular system for pairing optimization, but with a very different set-partitioning algorithm. The Carmen System, named after Carmen Systems AB of Sweden and employed by many European airlines, includes a user graphical interface, an optimizer, a rule compiler, and a report generator. The user interface allows easy adjustment of rules and other data and thus facilitates "what-if" analysis. The optimization module removes infeasible pairings in a preprocessing step, and then follows Wedelin's algorithm (Wedelin, 1995), which uses Lagrangian relaxation and a coordinate search approximation instead of the more common but often less efficient branching methods. Mingozzi et al. (1999) also approach the set-partitioning problem differently for generalized crew scheduling, in an example from the crew scheduling literature beyond airline applications. Their algorithm finds a lower bound for the crew scheduling problem by heuristically solving the dual LP relaxation without the set-partitioning matrix.

Klabjan et al. (2002) use the LP-based branch-and-bound method they presented in 2001 (Klabjan et al., 2001) to solve the daily crew pairing problem, but generate columns based on a novel set of rules. They assert that scheduling crews before aircraft routings improves crew scheduling flexibility, as takeoff and landing times can be adjusted within short time windows without reducing the scheduled capacity on each route. They use plane-count constraints to maintain routing feasibility while minimizing crew costs, and therefore integrate the routing and crew scheduling problems somewhat. Plane-count constraints simply require that the flight flow across a specific time, usually time 0, does not exceed the available number of aircraft.

The crew pairing algorithms discussed so far have treated duty periods as subsets of pairings, such that the pairing cost is represented as the sum of duty period costs. This approach omits the lodging and ground transportation costs of overnight stays between duty periods in a pairing. To account for the total pairing cost, Vance et al. (1997) decompose the problem in two steps, first linking flight legs into duty periods, and then combining duty periods into pairings. Dantzig-Wolfe decomposition and column generation methods are used to solve the subproblems. While the solution provides a better linear programming bound, it is more difficult to solve than the conventional pairing set partitioning problem.

Rostering. Rostering is the development of monthly work schedules for crews. As discussed earlier, rosters may be personalized before or after optimization, depending on each airline's policy. The main objective of rostering is minimization of cost. However, these models may also be designed to maximize crew utilization and thus minimize the number of crews employed, since each crewmember has a fixed cost of benefits and minimum guaranteed pay. Therefore, some airlines can use rostering methods to help determine how many expensive crews to employ.

Ryan (1992) discusses a generalized set-partitioning algorithm for rostering that creates rosters that are more fair to crewmembers of all seniority levels, accounting for training requirements and decreasing the inequity in bidlining that results when only senior crews always get their preferences. He generates all possible work schedules for each crew member that follow collective bargaining rules and account for desiderata and appointments. The objective function accounts for the minimum cost sought by the airline as well as aircrew preferences, such as fewer trips in a month. He uses LP and constraint branching to solve the problem, partitioning the large number of columns using pricing schemes and countering degeneracy with Wolfe's method.

The algorithms presented by Gamache et al. (1999) construct personalized crew rosters for Air France, using column generation, set partitioning, and heuristics to find a good integer solution. This solution maximizes the total work time for each crewmember, within the labor rules. They then present approximations that significantly reduce computational time. Gamache and Soumis (1998) also approach Air France's rostering problem by moving desiderata constraints into the objective function and assigning penalties for violation, weighted according to the seniority of the employee requesting the time off. This method minimizes the required number of crew members, but does not satisfy all desiderata. The authors suggest specifying a minimum number of crew members in order to satisfy more desiderata. They also note that appointments can be moved from the preprocessing step into the optimization step in the same way.

Planning for disruptions. Most of the methods discussed so far are deterministic and result in schedules which can be difficult to recover when disruptions occur, as cost minimization leads to tighter connections, for example. A few investigators have proposed methods to plan for disruptions. Yen and Birge (2000) present a stochastic integer programming model that accounts for the cost of delays. They use a branching algorithm to identify connections with high total costs and select better alternatives. The algorithm avoids crew plane changes, thus reducing the chance of missed crew connections. Schaefer et al. (2003) use search algorithms with penalties and a set of assumptions to identify crew pairings with low costs when disrupted. They approximate the lower bound for optimal crew schedule costs with disruptions, and show that some of the crew schedules from their algorithm perform close to this lower bound. Both groups of investigators highlight the need for more work in this area, since deterministically optimal crew schedules can become very expensive when disrupted.

The operational problem. Plans rarely survive contact with reality. The very experienced personnel who work in airline operations control centers make decisions to minimize the impact of schedule disruptions that occur from a few days before departure through the time that a flight arrives. Since the objective of operations control is to get planes in the air and passengers to their destinations, cost concerns are a lower priority than in planning, and cost savings from optimization suffer. Stojković et al. (1998) propose minimizing the impact of disruptions on the crew schedule by considering a small subset of pairings chosen from those that are impacted by the disruption or are near enough in time and space to be potentially useful in recovery, and then generating and pricing recovery options from these pairings. They aim to restrict pairing changes to one per candidate crew. For larger disruptions, they propose selecting a longer period and more candidate pairings, and decomposing the pairing subset as needed to reduce computational time. Like Stojković et al., Lettovský et al. (2000) select candidate pairings for recovering disruptions, but they use a more flexible method of separating these candidates into individual flight legs. They then reward the reassignment of flight legs to the originally scheduled crew while searching for a minimum cost reassignment, maintain feasibility with crew flow conservation constraints. Note that both of these methods attempt to optimally recover limited disruptions resulting from crew delays or unplanned maintenance events. Methods for recovering major disruptions, due to large-scale events such as blizzards or thunderstorms, still require more investigation.

2.2.5 Maintenance scheduling

Just as crews require periodic training, aircraft require periodic maintenance, but maintenance is much more frequent and varied than crew training. Each aircraft must meet FAA and airline requirements for four different levels of scheduled maintenance, which can be planned. For example, Type A maintenance consists of visual inspections and minor work performed every three to four days, while Type D involves a major overhaul of all or part of an aircraft every six to twelve months. Most airlines have several *maintenance stations* where technicians perform scheduled maintenance. A *maintenance opportunity* occurs when an airplane spends a night at a maintenance station, whether or not maintenance is actually performed. A *routing* is a multi-day schedule for a single airplane, similar to a crew pairing. Obviously, maintenance opportunities must be built into aircraft routings, and aircraft can most easily begin, end, and change routings at a maintenance station. Note that some airlines plan routings by fleet type well in advance, while others include aircraft routing by specific airplane in current operations, planning each jet's route a few days before departure. The former approach allows airlines to better control each jet's exposure to varied weather conditions and other stresses and thus to distribute stress evenly across the fleet, which tends to reduce variability in major maintenance frequency. The latter method increases flexibility in responding to changes in aircraft availability, and often permits schedule recovery before customers are impacted.

Unscheduled maintenance occurs when an airplane "breaks" during a trip and requires immediate attention to remain in service after landing. These problems range from burned-out lights to engines damaged by birds, and generally must be fixed at the airport where the plane landed. Sometimes technicians or parts must be flown to non-maintenance stations to repair the plane, which takes time and exacerbates schedule disruption. While scheduled maintenance is largely predictable and deterministic, unscheduled maintenance occurs randomly. Thorough data collection may lead to useful stochastic models for developing schedules more robust to expensive unscheduled maintenance disruptions.

The research usually treats maintenance as a side or subproblem to other problems; only a few efforts specifically address the maintenance routing problem in its own right. For example, Talluri (1998) and Gopalan and Talluri (1998) developed procedures for routing aircraft to meet type A maintenance requirements as well as *balance checks*, which are less frequent, more expensive events performed at a single designated maintenance station (in the case of most airlines.) Their method heuristically finds maintenance-feasible four-day aircraft routings in polynomial time, and can identify flights which must be shifted to another fleet to achieve maintenance feasibility. This fast algorithm can be particularly useful when airlines route specific airplanes late in the planning process. However, Cohn and Barnhart (2003) show that by planning maintenance routings before crew pairings, the solution quality of both problems improves. They state that crew pairings often produce infeasible maintenance routings, and planning the maintenance routings first limits selected crew pairings to those which allow for required maintenance opportunities. Using a model that limits crew pairings to only maintenance-feasible legs and column generation solution techniques, they show that this approach improves upon the optimality gap of a basic integrated method that solves both problems in one LP.

Cohn and Barnhart's approach to the maintenance routing problem exemplifies efforts to integrate the subproblems of airline optimization. Several other efforts to integrate different aspects of this large problem have also recently been developed.

2.2.6 Subproblem integration efforts

The immense size of the airline scheduling problem prevents simultaneous solution of the entire problem in a single model. An airline may operate 2,500 flights in a day using 800 crews and visiting 200 airports with 10 fleet types. Given the number of possible flight times per day and side constraints such as maintenance and training requirements, the number of potential schedules easily reaches the billions. Currently, no methods exist for solving the entire scheduling problem with a single model. Therefore, efforts to solve multiple aspects of the problem together and thereby improve the solution for all subproblems involved usually involve decomposition or other iterative techniques. Most documented subproblem integration methods consider the crew scheduling and aircraft routing problems together for a predetermined schedule, like the method of Cohn and Barnhart (2003), or the more complex problem pair of flight and crew scheduling. Maintenance scheduling is often treated as a subset of routing. No integration efforts have yet progressed beyond the current sequential nature of the airline scheduling problem.

Cordeau et al. (2001) integrate the aircraft routing and crew scheduling problems for a single fleet type, thus sidestepping the fleeting problem. They apply Benders decomposition and partition the simultaneous scheduling problem into side constraints specific to each subproblem, such as FAA daily flight time limits for the crew and maintenance requirements for aircraft, and linking constraints which involve variables from both subproblems. While the Benders approach did not reduce computational time as compared to a column generation method without the decomposition, the optimality gap decreased by a factor of as much as 3 for a range of test problems.

Instead of fixing the flight schedule and varying routings as in Cohn and Barnhart (2003) and Cordeau et al. (2001), Stojković and Soumis (2001) fix the aircraft itineraries and adjust the flight times and crew pairings. They employ Dantzig-Wolfe decomposition and branch-and-bound methods and achieve computational times of only seconds or less in realistic test cases. The master problem contains constraints that cover the flights with crews and flight precedent constraints to maintain itinerary sequences, while the subproblems are time-constrained shortest path problems. Freling et al. (2000) attack the integrated crew and vehicle scheduling problem with Lagrangian relaxation, column generation, and heuristics rather than decomposition, and also achieve short computational times in their experiments. Their results allowed reduction of the number of drivers required in a vehicle scheduling problem.

While some general transportation optimization techniques, such as the work of Freling et al. (2000), can be applied to airline optimization, the constraints are quite different when frequent stops are possible versus when stops are very expensive. While bus passengers expect the bus to stop many times before reaching a destination, airplane passengers generally expect to reach their destination with as few stops or plane changes as possible. Ground-transported cargo is even less sensitive than bus passengers. Integrated scheduling techniques are more advanced for general transportation than for airline optimization, but must be applied to airline problems with caution because of the diverse nature of the problems.

One of the best efforts yet for integrating airline scheduling is proposed by Barnhart et al. (1998). The method uses a master problem consisting of the FAM and a duty pairing problem (DPP) model, which is an approximation of the crew pairing problem. The integrated model appears as follows, where constraints (2.3e), (2.3f), (2.3i), and (2.3j) correspond to the DPP. All data and variables are as seen in the FAM above, with the addition of node set L^k for the duty network of fleet k, duty elapsed time cost C_j, ground arc elapsed time cost D_h. Also, A_{ij} is 1 if duty j covers flight i, B_{lj} is 1 if duty j arrives at node l and -1 if it departs from node l, B_{hj} is the ground arc equivalent. The decision variables are X_j^k, which is 1 if duty j of fleek k is assigned to a crew and 0 otherwise, and Y_h^k, the number of fleet k crews on ground arc h.

$$\min \sum_{k\in K}\sum_{i\in F} c_{ik}x_{ik} + \sum_{k\in K}\sum_{j\in J} C_j^k X_j^k + \sum_{k\in K}\sum_{h\in H} D_h^k Y_h^k \tag{2.3a}$$

$$\text{s.t.} \quad \sum_{k\in K} x_{ik} = 1 \qquad \forall\, i \in F \tag{2.3b}$$

$$\sum_{i\in F} b1_{nik}x_{ik} + \sum_{h\in H^k} b2_{nhk}w_{hk} = 0 \qquad \forall\, n \in M^k, k \in K \tag{2.3c}$$

$$\sum_{i\in F} d1_{ik}x_{ik} + \sum_{h\in H^k} d2_{hk}w_{hk} \leq N_k \qquad \forall\, k \in K \tag{2.3d}$$

$$x_{ik} - \sum_{j\in J} A_{ij}X_j^k = 0 \qquad \forall\, i \in F, k \in K \tag{2.3e}$$

$$\sum_{j\in J} B_{lj}^k + \sum_{h\in H^k} B_{lh}^k Y_h^k = 0 \qquad \forall\, l \in L^k, k \in K \tag{2.3f}$$

$$x_{ik} \in \{0,1\} \qquad \forall\, i \in F, k \in K \tag{2.3g}$$

$$w_{hk} \geq 0 \qquad \forall\, k \in H^k, k \in K \tag{2.3h}$$

$$X_j^k \in \{0,1\} \qquad \forall\, j \in J, k \in K \tag{2.3i}$$

$$Y_h^k \geq 0 \qquad \forall\, h \in H^k, k \in K \tag{2.3j}$$

The fleet assignments from this integrated model are then entered into the crew pairing problem, which is solved to find a feasible, minimum-cost solution for the given fleet assignments. Although this method is sequential, the inclusion of the crew pairing problem approximation in solving the FAM can considerably improve the quality of the FAM solution with respect to crew costs, since the FAM alone may produce a result with an expensive minimum-cost crew pairing solution.

2.3 Summary

Numerous methods exist for solving network problems, including decomposition, column generation and pricing, and heuristic algorithms. These techniques have been applied to many facets of airline planning and operations individually, and the literature contains a few efforts at optimizing two or more pieces of the airline puzzle simultaneously. However, no method to integrate airline planning while accounting for the turbulence of operations has yet been proposed. The sequential nature of airline planning leads to a loss in optimality and therefore profit, so a method to account for each part of the problem in initial planning could improve the overall success of an airline considerably.

CHAPTER 3

The Airline Flight Schedule

The central problem of airline planning is the flight schedule. All other aspects of airline operations, including maintenance and crew scheduling, are derived from the master flight schedule, and may be considered as subproblems to it. Sequential methods of airline scheduling, in which a series of problems are solved sequentially after the flight schedule has been determined, lead to inherently suboptimal solutions, since small adjustments to accommodate one subproblem may have an unprofitable ripple effect in other subproblems. For example, a small schedule change made to minimize crew costs may lead to a large increase in fuel costs. To maximize overall profit, an integrated view of the master schedule and subproblems such as crew scheduling is required. Therefore, we begin by describing the master schedule as a dynamic network.

3.1 Master schedule

In the dynamic network representation of the master schedule, each node corresponds to a time and location. Nodes are connected by flight arcs or ground arcs which are directed in time. To describe the rules that govern which arcs exist between nodes, we review the basic operation of a flight network.

Flight networks consist of airports, and flights between the airports. Each airport services a nonnegative integer number of passengers who want to travel to other locations on flights. Flights require time to travel from one airport to another, and airplanes must spend a minimum amount of time at the arrival airport for reloading and cleaning before taking off again. Airports can service a limited number of planes at any time. Each airport has a finite gate capacity, as well as a limited number of

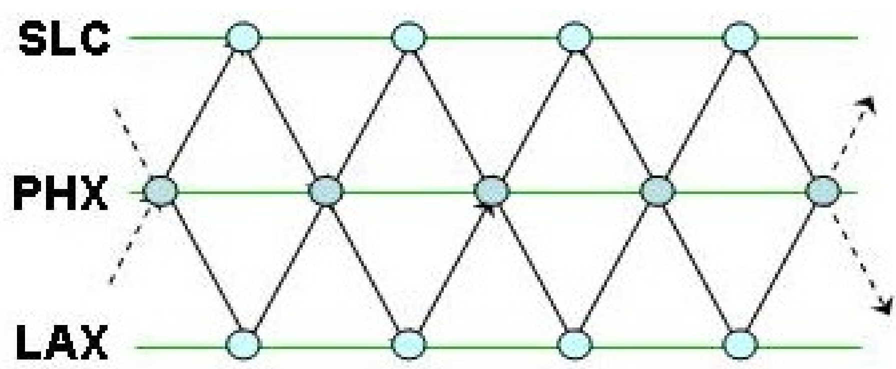

Figure 3.1: Notional time-space flight network

takeoff and landing time "slots" in any hour. Finally, a finite number of airplanes exist in any such system. Figure 3.1 displays a notional network with three cities, where time progresses horizontally, and the connecting lines represent flights and ground arcs.

The following model is similar to the Fleet Assignment Model (FAM) of Hane et al. (1995). Let I denote the set of locations or airports in an airline network, and T denote the set of discrete time increments, where $T = \{0, \ldots, c, \ldots, t_{max}\}$ and c is the "cycle time" of the system. A *cycle* is the length of the planned schedule, such that the schedule is repeated once per cycle. Typically, a cycle represents one day, and t_{max} may be any time up to one month. The standard time required to travel between locations i and j is t_{ij}, and the minimum turn time at location j is $\gamma(j)$, where an aircraft "turn" is the process of unloading passengers and cargo, cleaning, fueling, inspecting, and reloading the airplane. Demand for travel from location i to location j, expressed as the number of flights per day, is d_{ij}. For simplification, we ignore the time-dependent nature of demand, such as lower demand for overnight flights. Coefficients c_{ij} denote the fuel cost for a flight from location i to location j.

Since each node in the dynamic network corresponds to a time and location, if $\mathcal{N}$ denotes the set of nodes, then $\mathcal{N} \subset I \times T$. For each $n \in \mathcal{N}$, $i(n)$ and $\tau(n)$ denote the location and time of the node, respectively. Let $\mathcal{A} \subset \mathcal{N} \times \mathcal{N}$ denote the set of arcs. If $(nm) \in \mathcal{A}$ corresponds to a scheduled flight, then $i(n) \neq i(m)$ and $\tau(n) = \tau(m) + t_{i(n)i(m)}$. If (nm) represents time spent on the ground, then $i(n) = i(m)$ and $\tau(n) = \tau(m) + 1$. We refer to these as *flight arcs* and *ground arcs*, respectively.

Let $\mathcal{N}(t) \subset \mathcal{N}$ denote the set of nodes associated with time t, i.e., $\mathcal{N}(t) = \{n \mid n \in \mathcal{N}, \tau(n) = t\}$. Similarly, let $\mathcal{A}(t) \subset \mathcal{A}$ denote the set of arcs that originate at time t, $\mathcal{A}(t) = \{(nm) \mid (nm) \in \mathcal{A}, n \in \mathcal{N}(t)\}$.

Each airport i is subject to time-invariant upper bounds on takeoff and landing rates, denoted as r_i^- and r_i^+, respectively. Further, no more than g_i airplanes may accumulate at airport i at any time. A total of V airplanes are available in the system. While an airline may own and lease more than V airplanes, at any time, some may be in long-term maintenance, retrofit, or otherwise not available for service; V captures this fact.

Let $\mathcal{B}$ be the set of subproblems, where $f_b(x)$ represents the objective value cost contribution for subproblem $b \in \mathcal{B}$. Subproblems include crew scheduling, fleeting, revenue generation, and maintenance scheduling. Note that the revenue generation subproblem's contribution to the objective function will appear as a negative "cost," driving the model to maximize revenue while minimizing other costs. The Master Schedule Problem (MSP) is as follows, beginning with a summary of the notation used in the model.

Sets:

I	Airport indices
T	Discrete time increment indices
$\mathcal{N}$	Location-time nodes, where $\mathcal{N} \subset T \times I$
$\mathcal{N}(t)$	Nodes that correspond to time t

$\mathcal{A}$	Arcs, where $\mathcal{A} \subset \mathcal{N} \times \mathcal{N}$
$\mathcal{A}(t)$	Arcs that are active at time t, i.e. $\tau(n) \leq t$ and $\tau(m) > t$ for $(nm) \in \mathcal{A}(t)$
$\mathcal{A}^{\mathcal{F}}$	Flight arcs, where $\tau(m) - \tau(n) = t_{i(n)i(m)}$, and $i(n) \neq i(m)$ for $(nm) \in \mathcal{A}^{\mathcal{F}} \subset \mathcal{A}$
$\mathcal{A}^{\mathcal{G}}$	Ground arcs, where $\tau(m) - \tau(n) = 1$, and $i(n) = i(m)$ for $(nm) \in \mathcal{A}^{\mathcal{G}} \subset \mathcal{A}$
$\mathcal{B}$	Subproblem indices

Data:

t_{max}	Index of final time increment
c	Cycle time, or length of one day
t_{ij}	Travel time from location i to location j
c_{ij}	Fuel cost for travel from i to j
$\gamma(i)$	Minimum turn time at location i
r_i^+, r_i^-	Maximum takeoff and landing rates at location i
g_i	Ground capacity of location i
V	Number of airplanes available in the system
$i(n)$	Location of node $n \in \mathcal{N}$
$\tau(n)$	Time of node $n \in \mathcal{N}$

Decision variables:

x_{nm}	Flow (i.e., number of aircraft scheduled) on arc $(nm) \in \mathcal{A}$

MSP:

$$\min \sum_{(nm)\in\mathcal{A}^{\mathcal{F}}} c_{i(n)i(m)} x_{nm} + \sum_{b\in\mathcal{B}} f_b(x) \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{m} x_{nm} \leq r^{-}_{i(n)} \quad \forall\, n \in \mathcal{N}, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{3.1b}$$

$$\sum_{n} x_{nm} \leq r^{+}_{i(m)} \quad \forall\, m \in \mathcal{N}, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{3.1c}$$

$$\sum_{n} x_{nm} \leq g_{i(m)} \quad \forall\, t, m \in \mathcal{N}(t), (nm) \in \mathcal{A}^{\mathcal{G}} \tag{3.1d}$$

$$\sum_{(nm)\in\mathcal{A}(0)} x_{nm} \leq V \tag{3.1e}$$

$$\sum_{(mn)\in\mathcal{A}^{\mathcal{F}}, \tau(p)-\gamma(i(n))<\tau(n)\leq\tau(p), i(n)=i(p)} x_{mn} \leq x_{pq} \quad \forall\, (pq) \in \mathcal{A}^{\mathcal{G}} \tag{3.1f}$$

$$\sum_{m} x_{nm} - \sum_{m} x_{mn} = 0 \quad \forall\, n \in \mathcal{N} \tag{3.1g}$$

$$x_{nm} - x_{n'm'} = 0 \quad \forall\, (nm) \in \mathcal{A}, i(n) = i(n'), i(m) = i(m'), \tau(n') = \tau(n) + c \tag{3.1h}$$

$$x_{nm} \in Z^{+} \quad \forall\, (nm) \in \mathcal{A} \tag{3.1i}$$

The objective function minimizes the fuel costs of the schedule plus the subproblem cost contributions. Constraints (3.1b) limit the number of flights departing a location to the takeoff capacity, while (3.1c) similarly limits the number of simultaneous landings at each location. To ensure that holding capacity is not exceeded at a location, we impose (3.1d). Constraint (3.1e) prescribes that the number of assigned airplanes does not exceed the number available, and is called a *plane count* constraint.

Equations (3.1f) require that flights stay on the ground for at least the minimum turn times, where $(pq) \in \mathcal{A}^{\mathcal{G}}$ is a ground arc. All of the flights that have arrived $\gamma(i(p))$ or fewer time units before $\tau(p)$ and are therefore in their mandatory ground hold time window are included in this constraint. Resource flow is maintained by (3.1g). Constraints (3.1h), known as *wraparound constraints*, ensure identical schedules for each cycle.

3.2 A note on fuel costs

As mentioned above, the MSP objective function accounts for fuel costs. Fuel costs typically comprise 12 to 15 percent of a major airline's operating costs, but may rise to 20 percent or more when oil prices rise (Reuters (2004), U.S. House of Representatives (2000), AP (2004)). For commuter airlines, fuel costs may be 25 percent of the total operating costs. Therefore, minimizing fuel costs is extremely important in maximizing an airline's profit.

The fluctuation in oil prices complicates the estimation of fuel costs for a schedule. While oil price forecasts for a three-month horizon are useful for planning, forecasts for several years in the future are less likely to be accurate. Therefore, careful consideration of fuel cost modeling is important in finding a profitable schedule. Although airlines should consider stochastic behavior in oil prices, weather, aircraft fuel burn rates, and any other factors that affect fuel burn rates, we defer investigation of fuel cost models for future research.

3.3 Surrogate models for subproblems

Given the model above, the other aspects of the airline scheduling problem can be represented as subproblems. We develop surrogate models for these subproblems and consider randomness in them when appropriate to improve the overall results found by MSP. While surrogate models will not yield exact changes in cost or feasibility of the subproblems, they can provide a means for finding the most profitable schedule for a given flight network.

3.4 Performance measures and solution procedure

To evaluate the impact of the surrogate models on the master schedule, we employ several performance measures. The most obvious performance metric is the objective value, and we can observe the relative change in objective value when each subproblem is considered. In the crew and revenue management surrogate models, we account for uncertainty as well, and employ the L-shaped method with numerous randomly generated scenarios. This will yield biased estimates of the objective function value (Mak et al. (1999)). In order to eliminate the biased imposed by our solution procedure, we use an independent sample with a sufficiently large number of observations of the random variable to ensure that our objective value estimates are accurate to within 1% with 95% confidence. We note that this level of accuracy in the estimated objective function value typically requires a much larger number of observations than was used in the solution procedure. Once the solution procedure terminates, we fix the master schedule solution. We then randomly generate independent scenarios in the subproblem and calculate a 95% confidence interval on the subproblem objective value after each scenario generation. When the width of the confidence interval is at most 1% of the estimated subproblem objective value, we report both the confidence interval and the percentage tolerance used to terminate the confidence interval calculation.

As the objective value of a surrogate subproblem is an approximate measure and not necessarily a true reflection of cost, it allows relative comparison of results, but the actual values have limited utility. Therefore, we compare flight schedule solutions as well, using measures that quantify the changes in the flight schedule resulting from consideration of the subproblems.

We compare numerous pairs of schedules resulting from the various computational experiments. For each pair of schedules, we count the number of flights that appear in both schedules, and consider the ensuing proportion. We also count the flights that are nearly the same in both schedules, defined as flights that share an O-D pair but fly two or fewer time increments later in the second schedule. Finally, when we consider schedule disruption with the crew scheduling subproblem, we count the consecutive ground arcs that exceed the minimum turn time in each schedule. We expect that a schedule with more time between flights is easier to recover following disruption than one with short scheduled turn times, and therefore use the count of "extended" ground arcs as a measure of schedule robustness. These counts are displayed in tables similar to the example seen in Table 3.1, where EGAs are extended ground arcs.

	Identical	"Near"	EGAs		Total Flights	
Fleet	flights	flights	Sch 1	Sch 2	Sch 1	Sch 2
1	15	8	12	18	52	53
2	7	14	15	24	49	54
Both	49	36	27	42	101	107

Table 3.1: Sample flight count summary.

For example, in Table 3.1, we see that as a result of considering the possibility of disruption, the total number of flights scheduled increased from 101 to 107. Both fleets experience an increase, although the increase is more notable for fleet 2 than for fleet 1. Nearly half of the scheduled flights for fleet 1 and about 40% of the flights for fleet 2 in the Schedule 2 are identical to, or the same as, in Schedule 1. We also observe that about 85% of the flights are identical or nearly the same when fleet assignment is ignored, so the flight service offered by the two schedules is quite similar. Further, Schedule 2 has a few extra flights, as well as more EGAs. This indicates that Schedule 2 has more slack to allow for recovery in the event of disruption, and that the extra considerations have shaped the schedule to include more flights.

In addition to the direct comparison of the flight schedules, we tally the flights in each schedule by origin, destination and departure *time window*, where a time window is a significant segment of the day such as morning (4am-12pm), afternoon (12pm-8pm), or night (8pm-4am). We can compare the flight tally for each O-D pair and time window of two different schedules, and identify differences in flight service between the two schedules. Note that we are tracking only nonstop flights and not ground arcs. Table 3.2 demonstrates a flight tally.

FLEET 1	**Schedule 1**					**Schedule 2**					**Difference**				
City	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
	"Morning"														
1		1	6	0	1		0	4	0	0		1	2	0	1
2	0		0	0	1	0		0	0	3	0		0	0	-2
3	3	0		1	0	1	1		0	2	2	-1		1	-2
4	1	0	0		0	1	0	0		0	0	0	0		0
5	1	0	2	0		2	0	3	0		-1	0	-1	0	

Table 3.2: Sample flight tally.

The flight tally tables for the experiments in the chapters which follow are rather extensive. Therefore, summary discussions of the tally results are included without the full set of tables. For example, in Table 3.2, we see that the flight service offered has changed by one or two flights between each city pair during this time window in most cases. These flights may be added, removed, or shifted to a different time window to increase overall profitability.

Given these flight schedule performance measures, we will proceed to develop surrogate representations of some operations subproblems that influence airlines schedules.

CHAPTER 4

Maintenance Scheduling: A Surrogate Representation

Just as crews require periodic training, aircraft require periodic maintenance, although maintenance requirements are more frequent and varied than crew training. Each US-operated aircraft must meet FAA and airline requirements for scheduled maintenance. Aircraft must undergo four types of maintenance, as explained by Talluri (1998), from infrequent major overhauls to minor visual inspections every few days. Major overhauls are generally accommodated on a rotating basis, and have the impact of an effective reduction in the number of aircraft available at any given time. Visual inspections require one or two hours every 65 flying hours by FAA rules, although most major airlines self-impose a stricter 45-hour maximum interval. Other types of minor maintenance include engine oil changes and other minor tasks needed to keep an airplane in service. Our model applies to consideration of the impact of minor maintenance on the flight schedule.

Current methods tend to consider maintenance schedules after the flight schedule has been determined. The reasons for this are fairly clear from a computational point of view. Scheduled maintenance requires knowledge of the flights assigned to each specific aircraft (called the "tail assignment"). The immense size of an airline's operations, involving up to 2500 flights, 200 airports and 10 distinct fleets, makes it unrealistic to expect to consider the tail assignment and maintenance schedule while initially planning the master schedule. From a decision modelling point of view, the reasons are equally clear. The master schedule, which forms the nucleus of the airline's business operations and revenue generating capacity, is the product of a strategic planning process in which big-picture issues such as competition and market shares are of primary importance. Maintenance, on the other hand, takes place on an operational level. As such, it requires a closer level of attention to detail.

Nonetheless, maintenance requirements impose constraints on the operation of the master schedule, which must be "maintenance feasible."

Talluri (1998) and Gopalan and Talluri (1998) developed procedures for routing aircraft to meet basic maintenance requirements. Their method heuristically finds maintenance-feasible four-day aircraft routings in polynomial time, and can identify flights which must be shifted to another fleet to achieve maintenance feasibility. This algorithm can be particularly useful when airlines route specific airplanes late in the planning process. Alternatively, Cohn and Barnhart (2003) show that by planning maintenance routings before crew pairings, the solution quality of both problems improves. They state that crew pairings often produce infeasible maintenance routings, and planning the maintenance routings first limits selected crew pairings to those which allow for required maintenance opportunities. Throughout our development, we assume that all minor maintenance can be performed at a set of properly equipped and staffed *maintenance stations*. A *maintenance opportunity* occurs when an airplane spends a sufficiently long period of time at a maintenance station, whether or not maintenance is actually performed. As a stand-in for the more complex notion of maintenance feasibility, our basic premise is that if a schedule has sufficient maintenance opportunities built into it, the schedule is likely to be maintainable with out excessive delay.

4.1 A model of maintenance opportunities

Minor frequent maintenance requires each aircraft to pass through a maintenance station every few days. As previously discussed, scheduling this maintenance results in a large integer program, which is prohibitive as a component of MSP. We incorporate this maintenance requirement by counting *maintenance opportunities*, layovers meeting or exceeding a minimum time interval at a designated maintenance station, within the MSP. Since minor maintenance must occur every $\mathcal{T}$ days (typically, $\mathcal{T}$ is 3-4 days), we can count the number of maintenance opportunities in this time window, and require that the number of maintenance opportunities be at least as large as the number of aircraft. To enforce this requirement, we add the following constraint to MSP, where we assume that $\mathcal{T} > 0$, and present a model from which $f_{mx}(x)$, the number of maintenance opportunities associated with master schedule x, can be derived.

$$f_{mx}(x) \geq \sum_{(nm) \in \mathcal{A}(0)} x_{nm} \tag{4.1}$$

The constraint (4.1) is equivalent to including the extended real value function $f^e_{mx}(x)$ in the MSP objective, where $f^e_{mx}(x) = 0$ if $f_{mx}(x) \geq V$, and $+\infty$ otherwise. Computationally, the constraint (4.1) is preferred.

Note that the new constraint (4.1) does not guarantee that each individual aircraft will be serviced, nor does it minimize maintenance cost; it simply offers a measure of whether a schedule is "maintainable." The maintenance opportunity count also acts as a measure for comparison between alternate schedules to estimate which is the most maintainable.

In the following discussions, we restrict our attention to the case in which there is only a single type of maintenance that must be scheduled, and comment on extensions to multiple types of maintenance at the end of the section. We begin by constructing a "maintenance network" based on the dynamic network used for MSP. To do this, we overlay "maintenance arcs" on the network, which correspond to sequences of enough ground arcs to permit maintenance to take place.

Figure 4.1 illustrates the flight network and the resulting maintenance opportunities. Figure 4.1a displays the basic MSP structure, where horizontal lines represent potential ground arcs, diagonal lines represent potential flights, and time moves from left to right. Phoenix (PHX) appears as a maintenance station, while Salt Lake City (SLC) and Los Angeles (LAX) do not have maintenance capability. Figure 4.1b depicts a collection of flights (and ground holds) between these cities. Assuming that maintenance requires two time periods and that at most two aircraft can undergo

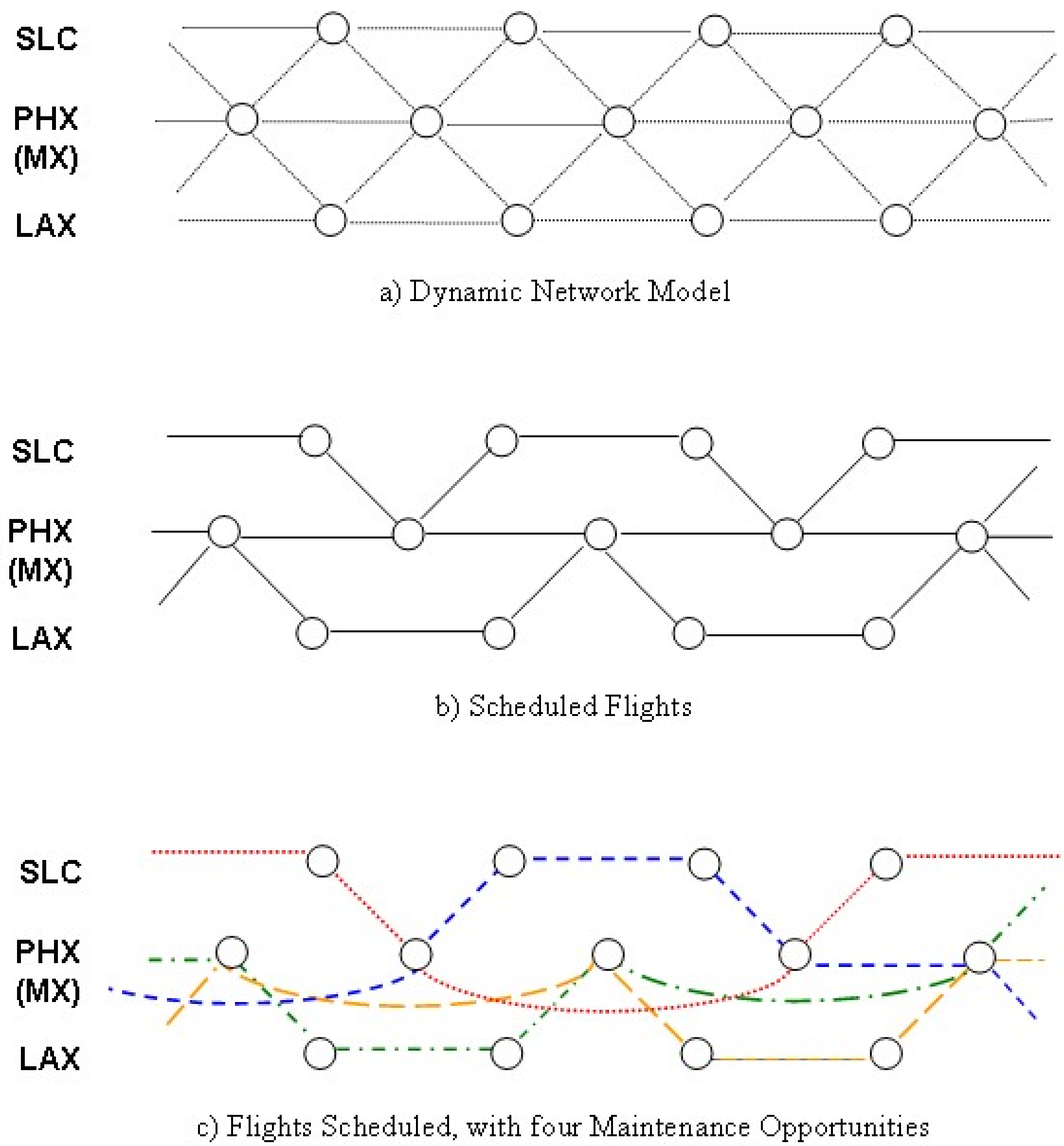

Figure 4.1: Network structure.

maintenance at any time, Figure 4.1c, illustrates a reorganization of the flights that provides four maintenance opportunities.

In addition to the notation established for MSP, we present the following definitions.

Sets:

I^{mx}	Maintenance stations, or the set of locations where scheduled maintenance can be performed
$\mathcal{N}^{mx}$	$\mathcal{N}^{mx} \subset \mathcal{N}$, where $n \in \mathcal{N}^{mx}$ if $i(n) \in I^{mx}$
$\mathcal{A}^{mx}$	Maintenance arcs, where $(nm) \in \mathcal{A}^{mx}$ if $i(m) = i(n), i(n) \in \mathcal{I}^{mx}$, and $\tau(m) - \tau(n) \geq t_{mx}$
$\mathcal{A}^{mx}(t)$	Maintenance arcs active at time t, i.e. $\{(n, m) \in \mathcal{A}^{mx} \mid \tau(n) \leq t$ and $\tau(m) > t\}$

Data:

t^{mx}	Time required to perform scheduled maintenance
h_i	Number of crews at maintenance station i
$\mathcal{T}$	Maintenance cycle time, typically three to four days

Decision variables:

y_{nm}	Flow on maintenance arc $(nm) \in \mathcal{A}^{mx}$
z_{nm}	Flow on non-maintenance arc $(nm) \in \mathcal{A}^{\mathcal{G}}$

The surrogate maintenance feasibility problem (MX) is as follows, given master schedule x from the MSP.

MX:

$$f_{mx}(x) = \max \sum_{(nm) \in \mathcal{A}^{mx}} y_{nm} \tag{4.2a}$$

$$\text{s.t.} \sum_{(\ell m) \in \mathcal{A}^{mx}(\tau(n))} y_{\ell m} \leq h_{i(n)} \quad \forall\, n \in \mathcal{N}^{mx} \tag{4.2b}$$

$$\sum_{(\ell m) \in \mathcal{A}^{mx}(\tau(n))} y_{\ell m} + z_{np} = x_{np} \quad \forall\, n, (np) \in \mathcal{A}^{\mathcal{G}}, i(n) \in \mathcal{I}^{mx} \tag{4.2c}$$

$$y_{nm} \geq 0 \quad \forall\, (nm) \in \mathcal{A}^{mx} \tag{4.2d}$$

$$z_{nm} \geq 0 \quad \forall\, (nm) \in \mathcal{A}^{\mathcal{G}} \tag{4.2e}$$

Though the goal of MX is to determine whether a schedule x is maintenance feasible, the objective is to maximize the flow on the maintenance arcs. While driving MSP to find a maintenance feasible solution, this objective may also provide a "cushion" of available maintenance arcs above the minimum number, which can make the master schedule more robust to disruptions that may reduce maintenance feasibility.

Constraints (4.2b) ensure that the number of maintenance opportunities does not exceed the number of available maintenance crews at any place and time. Equations (4.2c) assign the flow on ground arcs at maintenance stations in the master schedule to maintenance or non-maintenance arcs.

Our presentation of the MSP-MX model highlights the potential to solve this problem using a variation of Benders' Decomposition. This can be readily accomplished. In cases where the flight network is only moderately sized, the maintenance constraints can simply be directly incorporated into the MSP without resorting to a decomposed presentation. In Section 4.4 we discuss extensions of the model, many of which favor solution by decomposition. We note that the use of the surrogate measure "maintenance opportunities" in place of the requirement of maintenance-feasibility eases the computational requirements of such integrated decision making.

Nonetheless, it simultaneously raises questions regarding the suitability of the surrogate. Ideally, a large number of maintenance opportunities would indicate a readily maintainable schedule, while a small number would indicate a schedule with the potential for maintenance difficulties. In other words, additional effort must be expended in order to determine the validity of the approximation.

4.2 Validation of the surrogate model

To test and demonstrate the validity of the surrogate model, we undertake the more involved task of actually testing the maintenance feasibility of the master schedule identified via the solution of the integrated model (MSP). This is accomplished via a "string model" similar to those described in Barnhart et al. (1998). A string is a series of flights that are assigned to a single aircraft. Although string models are not the most efficient method for solving the maintenance scheduling model, they are relatively simple and provide a useful basis for comparison with the approximation results. The heuristic method proposed by Talluri (1998) is more efficient computationally, but is more difficult to implement. Our goal in this section is to explore the validity of the output of our surrogate model rather than to efficiently identify an operationally viable maintenance schedule. For this reason, we validate the surrogate representation with the more easily implemented, but computationally time consuming, string model.

Our validation model begins with the a collection of binary variables.

Binary decision variables:

f_{nms} 1 if flow on arc $(nm) \in \mathcal{A}$ is assigned to string s; 0 otherwise

ϕ_{ns} 1 if a maintenance arc begins at node n on string s; 0 otherwise

σ_s 1 if string s contains at least one maintenance arc; 0 otherwise

In this manner, $\sum_s \sigma_s$ denotes the number of aircraft that utilize at least one maintenance arc. If this number is at least as great as the number of aircraft needed to fly the schedule, then the schedule is "maintenance feasible." If the maximum number of strings that contain maintenance arcs is less than the number of aircraft, then the schedule is not maintenance feasible.

To represent potential maintenance arcs and minimum ground holds, we introduce two sets. These sets are primarily notational aids for expressing the model. For each n such that $i(n) \in \mathcal{I}^{mx}$, let

$\mathcal{A}^{mx}_n$ $=\{(\ell, m) \in \mathcal{A}^{mx} \mid i(\ell) = i(n),\ \text{and} \tau(\ell) = \tau(n)\}$, the set of maintenance arcs that originate with node n.

$\mathcal{A}^{\mathcal{G}}{}_n$ $=\{(\ell, m) \in \mathcal{A}^{\mathcal{G}} \mid i(\ell) = i(n), \tau(n) \leq \tau(\ell) < \tau(n) + \gamma_i(n)\}$., the set of ground arcs necessary to satisfy the minimum ground hold constraint for flights landing at n.

The set $\mathcal{A}^{mx}_n$ represents the arcs that may be used to schedule maintenance at location $i(n) \in \mathcal{I}^{mx}$ beginning at time $t(n)$. Similarly, the set $\mathcal{A}^{\mathcal{G}}{}_n$ represents the set of ground arcs that are associated with the minimum ground hold constraint for flights landing at location $i(n)$ at time $t(n)$.

Given a master schedule x, a more precise representation of a maintenance schedule may be obtained via the solution to

$$\max \sum_s \sigma_s \tag{4.3a}$$

$$\text{s.t.} \quad \sum_{n \in \mathcal{I}^{mx}} \phi_{ns} \geq \sigma_s \qquad \forall\, s \tag{4.3b}$$

$$\sum_{(\ell m) \in \mathcal{A}_n^{mx}} f_{\ell m s} \geq \phi_{ns} \qquad \forall\, n \in \mathcal{N}^{mx}, s \tag{4.3c}$$

$$\sum_s \sum_{n: i(n)=i,\ \tau(n) \leq t < \tau(n)+t^{mx}} \phi_{ns} \leq h_i \qquad \forall\, t,\ \forall\, i \in \mathcal{I}^{mx} \tag{4.3d}$$

$$\sum_m f_{mns} = \sum_m f_{nms} \qquad \forall\, n \in \mathcal{N}, s \tag{4.3e}$$

$$\sum_s f_{nms} = x_{nm} \qquad \forall\, (nm) \in \mathcal{A} \tag{4.3f}$$

$$\sum_{m:(mn) \in \mathcal{A}^{\mathcal{F}}} f_{mns} \geq \sum_{(\ell p) \in \mathcal{A}^{\mathcal{G}}{}_n} f_{\ell s} \qquad \forall\, n \in \mathcal{N}, s \tag{4.3g}$$

$$\sigma_s, \phi_{ns}, f_{nms} \in [0,1] \qquad \forall\ n, m, s \tag{4.3h}$$

Constraints (4.3b) and (4.3c) drive variables σ_s to indicate whether string s contains a maintenance arc or not. Since σ_s is binary, the objective is increased by 1 if one or more maintenance arcs is assigned to a string, so that assigning more than one maintenance arc to a string does not result in any additional benefit. In (4.3d), flow on maintenance arcs at any one place and time is limited to the number of available crews, as in (4.2b). Flow is conserved by (4.3e). These constraints also associate individual flight legs with strings. Equations (4.3f) require that each unit of flow on an arc is assigned to a string, and (4.3g) ensure that landing flights enter a ground hold or maintenance arc. In a maintainable flight schedule, every string will include at least one maintenance opportunity, so that $\sum_s \sigma_s$ equals the total number of aircraft.

The string model follows the same maintenance time requirement as the approximation model. The solution obtained from the approximation model indicates whether the number of maintenance opportunities in the schedule is at least as great as the number of aircraft. Consequently, the string model cannot find more maintenance arcs than the surrogate model. If the surrogate model indicates that not enough maintenance arcs exist for the schedule to be maintainable, it is impossible for the string model to show a contradictory result. When the surrogate model indicates maintenance feasibility, it is not immediately clear that the schedule is necessarily maintainable. We will test this empirically.

4.3 Experiment

Using the string model, we now have a basis from which to test the validity of our surrogate representation of maintenance. Within our experimental setting, we used "small" networks containing five airports and 144 thirty-minute time periods, or three days, with a maximum of 15 aircraft in each schedule. The sizes of the test problems are listed in Appendix B. Each of the problems that we used was constructed using randomly generated data, as described in detail in Appendix C. Flight schedules were generated for each network. Maintenance feasibility of these schedules was evaluated using both the surrogate model and the string model. The results were then compared for consistency. Our goal is to test the predictive capability of the surrogate model for a variety of master schedules on these various networks. If the string model finds a problem to have fewer maintainable strings than the number of aircraft needed to fly the schedule, then we say that the schedule is not maintainable. Similarly, if the surrogate model finds fewer maintenance opportunities than aircraft, the schedule is identified as unmaintainable. For each of the networks generated, we compared the indications of maintenance feasibility obtained from the surrogate model to those obtained from the string model.

Table 4.1 contains a summary of these comparisons for five different sets of problem data. MSP was used to generate flight schedules. Since $f_{mx}(x)$ in constraint (4.1) is initially zero, the constraint on maintenance opportunities, (4.1), if effectively inactive for the first schedule x generated. We use "a" to indicate an MSP solution based solely on profit (i.e., with (4.1) omitted), and "b" to indicate the profit maximizing solution with (4.1) included. Consequently, in the first data set, profit-only considerations yield six maintenance opportunities, and results in six maintainable flight strings for 15 airplanes. Adding the constraint on maintenance opportunities results in an abundance of maintenance opportunities (120 opportunities for 15 airplanes), which the string model confirms as maintainable. In all cases, the surrogate and the string model reached the same conclusion regarding maintenance feasibility. The results of this experiment indicate that the surrogate model is useful for determining whether a schedule is maintainable.

Problem Number	**# Planes**	**Surrogate Model MX opportunities**	**String Model # maintained**
1a	15	6	6
1b	15	120	15
2a	15	12	9-12
2b	14	36	14
3a	15	6	6
3b	15	84	15
4a	15	12	0
4b	15	30	15
5a	15	6	6
5b	15	30	15

Table 4.1: Results of model validation experiment.

We note that in each case, the MSP objective value was not affected by the inclusion of the constraint on maintenance opportunities.

As described in Section 3.4, the Table 4.2 contains the following metrics for comparing two schedules:

- The number of flights that appear in both schedules
- The number of flights that are nearly the same in both schedules, defined as flights that share an O-D pair but fly two or fewer time increments later in the second schedule

- The total number of flights in each schedule

For each test case, the most profitable schedule for the initial master schedule (Init) is compared with that with maintenance considerations (Mx). All test cases include a single fleet. As shown in Table 4.2, the vast majority of flights in each test

Test Case	Identical flights	"Near" flights	Total Flights Init	Total Flights Mx
1	2	7	110	40
2	0	9	42	44
3	2	5	90	30
4	4	6	50	56
5	10	8	112	24

Table 4.2: Flight count summary.

case were scheduled differently under consideration of maintenance feasibility. This indicates that the algorithm explored alternate optimal solutions as the Benders' cuts shaped the master schedule towards a maintenance-feasible one.

As expected, the solution times of the two maintenance models differed sharply. The time required to solve the surrogate model is essentially negligible, while the string model requires much more time to solve, even for these small networks. In some cases (e.g., 2a), the string model failed to identify an optimal solution after several hours (although it indicated a lower bound of 9 maintainable strings and an upper bound of 12). When employing a solution technique that requires the subproblems to be solved many times, such as Benders' decomposition, these time savings are extremely helpful in finding an overall optimal schedule within a "reasonable" amount of time.

To demonstrate scalability, we applied the master schedule planning process to much larger networks, with 20, 25, and 30 airports and 144 time periods. The resulting problems are too large to subject the schedule to an investigation of maintenance feasibility via our string model. However, in each of the problems that we generated and solved, we found that the solutions obtained using the MSP with the constraint on maintenance included sufficient opportunities for the aircraft to be maintained, whereas the initial solutions for the same problems were generally not maintainable schedules. As in the smaller problems, the objective values were not affected by the inclusion of the maintenance constraint. Table 4.3 shows the results of solving a few large problems with and without maintenance considerations in the model.

Problem Number	**# Airports**	**# Planes**	**Initial # MX opportunities**	**Final # MX opportunities**
6	20	60	0	66
7	25	75	6	92
8	30	90	60	112
9	30	90	0	112
10	30	90	18	108
11	30	90	24	109

Table 4.3: Demonstration of scalability.

4.4 Extensions

The discussion above explores the validity of a surrogate model in approximating maintenance scheduling, and thus to rapidly judge a schedule for maintenance feasibility. For simplicity, the MX model did not consider several factors that impact real aircraft maintenance scheduling. Some of these considerations may be explored as extensions of our model.

"Safety factor." In our model, we required that the number of maintenance opportunities be at least as great as the number of aircraft needed to fly the schedule. Since the number of aircraft available may vary, especially for larger networks, and since delays and other schedule disruptions may cause deviations from the planned schedule, we may want to include a multiplier on V in the model. For example, we may want the schedule to contain at least $\alpha > 1$ maintenance opportunities per plane, so the right-hand side of constraint (4.1) becomes $\alpha \sum_{(nm)\in\mathcal{A}(0)} x_{nm}$. Inclusion of this safety factor may decrease the profit of the schedule, so the tradeoffs between profit and reduced risk of loss of maintenance feasibility should be analyzed.

Multiple forms of maintenance. Our investigation assumed a single type of aircraft maintenance. In reality, different types of maintenance must be performed at different intervals, and possibly in different locations. Major maintenance, such as annual inspections, may be modeled by reducing V, reflecting the fact that a given number of aircraft are routinely removed from service for an extended period of time. With varying forms of frequent maintenance required, we may use multiple sets of maintenance arcs, i.e., $\{\mathcal{A}_i^{mx}\}$, and constrain each type separately.

Random maintenance times. When mechanics discover problems on an aircraft during a short inspection, repair times can easily exceed t^{mx}. To capture scheduled maintenance events that exceed the predicted time, we may reformulate MX as a stochastic linear program with randomness in the number of time periods required for membership of (nm) in set A^{mx}. Since the number of potential scenarios for extended maintenance can be large, our solution technique would need to incorporate stochastic programming solution techniques, which typically favor the decomposed presentation of the integrated model.

Unscheduled maintenance. Unscheduled maintenance occurs when an airplane "breaks" during a trip and requires immediate attention to remain in service after landing. These problems range from burned-out lights to engines damaged by birds, and generally must be fixed at the airport where the plane landed. Sometimes technicians or parts must be flown to non-maintenance stations to repair the plane, which takes time and exacerbates schedule disruption. While scheduled maintenance is largely predictable, unscheduled maintenance occurs randomly. While our focus in this chapter is on scheduled maintenance, the impact of schedule disruptions on the operation of the flight schedule is examined in the context of crew scheduling.

4.5 Conclusions

As discussed, airline schedules are very large, complex problems which currently are not typically developed in a manner that explicitly recognizes the impact of operational problems on the profitability of the schedule. Approximations of these related operational problems can allow the development of a schedule based on a more realistic representation of its ultimate profitability. Our validation experiment demonstrates the viability of one such approximation model. In particular, it appears that the approximation model is a useful surrogate for the exact model, and can be used for indicating maintenance feasibility of flight schedules.

The data and models used in this demonstration are simplified. For example, the models do not account for common requirements such as overnight layovers. Similarly, the profit-per-flight data depend only on the origin and destination of the flight, ignoring the influence of time of day, crew costs, winds, and myriad other factors. Many of these items can be incorporated in a more comprehensive version of MSP. Nevertheless, the concept of the surrogate model as demonstrated is still effective.

The surrogate model for maintenance is clearly useful in identifying feasible schedules. The model may also be used for long-term planning decisions, such as how many maintenance stations to operate and where to locate them. These types of problems can be analyzed by adding relative cost coefficients for the maintenance stations to the objective function, and by changing the maintenance crew availability data to reflect each option.

While we have not considered of randomness in the maintenance subproblem here, the techniques presented in the sections that follow may be applied to the maintenance subproblem as well. Stochastic models can lead to schedules that are more likely to remain maintenance feasible even when operations do not flow as planned.

CHAPTER 5

Crew Cost Approximation

Crews are an expensive and essential component of airline operations. As discussed in Chapter 2, extensive research on decision models to support crew scheduling has been conducted, and has primarily focused on optimally assigning crews to a predetermined flight schedule. Some methods allow small adjustments to the flight schedule to lower crew costs or maintain feasibility, but major changes are generally not considered. Models that consider the tradeoffs between the master schedule and the crew schedule have not been examined, probably because the crew scheduling problem is very time-consuming to solve and numerous iterations are thus impractical.

Crews incur costs at several stages of their trips. Most obviously, they must be paid for block time, as defined in Chapter 2. They also receive pay for sit time and deadhead legs. Overnight stays vary in cost depending on the location, and include transportation to and from the hotel, meal allowances, and lodging costs. We will assume that overhead costs such as medical benefits depend only on the number of crewmembers employed, and are constant among all schedules. Also note that we are generalizing the type of crew. Pilots and first officers, also called *technical crew*, earn considerably more money than flight attendants,who are subject to different FAA rules. Therefore, our use of the term “crew” primarily refers to technical crew, although the model could be expanded or generalized to include flight attendant costs as well.

In order to account for crew costs in planning the master schedule, we will develop an approximation model of crew schedule costs that overlays a crew duty network on the MSP solution. While this surrogate may not produce an exact figure of the true cost of an optimal crew schedule, it provides a basis for comparison with other candidate schedules. The crew duty network accounts for FAA rules such as maximum duty time per day and crew rest, and is used to estimate the crew costs for a schedule by summing the total costs for sit time, block time, deadheading, and overnights. Variation in crew pay rates will also be discussed.

5.1 Surrogate crew scheduling problem

The surrogate crew scheduling problem (SCSP) constraints model a network that is overlaid on the MSP flight network, like the surrogate maintenance model. In this network, one unit of flow corresponds to one crew. The model assigns crew flow for block time to scheduled flights, and connects the block time flow with sit time, deadheading, and overnights. According to FAA rules, a crew duty period may last no more than 12 hours. In the model, the total duty flow at any time may not exceed the total overnight flow 12 hours later. Overnight arcs must meet minimum crew rest requirements unless they occur at a pilot base. We will assume that overnights at pilot bases denote a change of crew, and therefore these arcs may be shorter than the FAA crew rest standard. In the model, t^{dp} is the duty time limit, and t^{cr} is the minimum crew rest time. We also impose an artificial minimum duty period length of 6 hours, denoted as t^{dm}, to improve crew utilization.

We will represent crew costs for arc $(nm) \in \mathcal{A}$ as c_{nm} with a superscript that identifies the cost component. Superscript f corresponds to flying time (block time or deadheading), o is overnight time, and s is sit time. We will use the term "sit time" to represent crew ground arcs, and will assume that the minimum crew turn time is less than or equal to one time period for the purposes of the approximation. Actual pay rates depend on the assigned crewmember's level of seniority, and thus crew pay rates are initially represented by an average cost. Most airlines in the United States currently pay the same rate for deadheading and block time. They also pay a guaranteed minimum rate for every hour of the duty day, so *sit* time, which is time spent briefing, debriefing, and waiting between flights, is paid at this rate. Overnight pay to the crew is only a few dollars per hour for personal expenses and the hotel bill, which varies by location, as described by Captain Mike Low of US Airways (2003). In addition to previously defined model constructs such as $\mathcal{A}^{\mathcal{F}}$, the set of flight arcs, we introduce additional model components that are specific to the SCSP. The crew cost approximation model based on a given schedule x is as follows.

Sets:

$\mathcal{A}^{\mathcal{F}}$	Flight arcs: $\mathcal{A}^{\mathcal{F}} \subset \mathcal{A}$, $i(n) \neq i(m)$, and $t(m) - t(n) = t_{i(n)i(m)}\ \forall\,(nm) \in \mathcal{A}^{\mathcal{F}}$
$\mathcal{A}^{\mathcal{G}}$	Ground arcs: $\mathcal{A}^{\mathcal{G}} \subset \mathcal{A}$, $i(n) = i(m)$, and $\tau(m) - \tau(n) = 1\ \forall\,(nm) \in \mathcal{A}^{\mathcal{G}}$
$\mathcal{A}^{\mathcal{D}}$	Duty arcs: $t^{dm} \leq \tau(m) - \tau(n) \leq t^{dp}\ \forall\,(nm) \in \mathcal{A}^{\mathcal{D}}$
$\mathcal{A}^{\mathcal{D}}(t)$	Duty arcs active at time t: $\mathcal{A}^{\mathcal{D}}(t) \subset \mathcal{A}^{\mathcal{D}}$, $\tau(n) \leq t$ and $\tau(m) > t$ $\forall\,(nm) \in \mathcal{A}^{\mathcal{D}}(t)$
$\mathcal{A}^{\mathcal{O}}$	Overnight arcs: $i(n) = i(m)$ and $\tau(m) - \tau(n) \geq t^{cr}\ \forall\,(nm) \in \mathcal{A}^{\mathcal{O}}$
$\mathcal{A}^{\mathcal{O}}(t)$	Overnight arcs active at time t: $\mathcal{A}^{\mathcal{O}}(t) \subset \mathcal{A}^{\mathcal{O}}$, $\tau(n) \leq t$ and $\tau(m) > t$ $\forall\,(nm) \in \mathcal{A}^{\mathcal{O}}(t)$
I^H	Crew base locations (H denotes "home"): $I^H \subset I$

Data:

c^f_{nm}	Crew cost for flying on leg $(nm) \in \mathcal{A}^{\mathcal{F}}$
c^s_{nm}	Crew cost for sit arc $(nm) \in \mathcal{A}^{\mathcal{G}}$
c^o_{nm}	Overnight cost for leg $(nm) \in \mathcal{A}^{\mathcal{O}}$
t^{dp}	Maximum total time per duty period (12 hours per FAA rules)
t^{dm}	Minimum total time per duty period (6 hours)
t^{cr}	Minimum crew rest time (8 hours per FAA rules)
C	Number of crews available to fly on any given day
H	Number of deadhead crews allowed on any flight

Input from MSP:

x_{nm} Schedule provided by the MSP

Decision variables:

y^b_{nm} block time flow on leg $(nm) \in \mathcal{A}^{\mathcal{F}}$

y^s_{nm} Sit time flow on leg $(nm) \in \mathcal{A}^{\mathcal{G}}$

y^d_{nm} Deadhead flow on $(nm) \in \mathcal{A}^{\mathcal{F}}$

y^o_{nm} Overnight flow on $(nm) \in \mathcal{A}^{\mathcal{O}}$

To connect overnight flow to crew duty flow and approximately enforce the 12-hour maximum duty period length rule, we use dummy variables that signify the beginning and end of a crew duty period. Without defining individual crew duty periods, we link the flow of crew from duty periods to overnight arcs via the dummy nodes. Therefore, we define the following variables.

Duty period variables:

δ_{mn} Flow of crew beginning a duty period at node m and ending at node n, $mn \in \mathcal{A}^{\mathcal{D}}$

The model that follows will thus construct duty period flow. A single crew duty period may appear as the one notionally depicted in Figure 5.1.

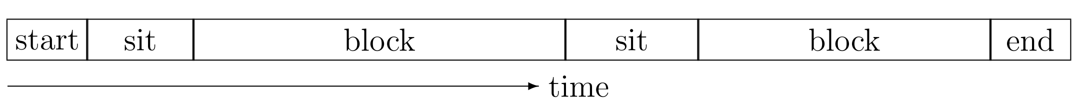

Figure 5.1: Notional duty periods.

$$f_{SCSP}(x) =$$

$$\min \sum_{(nm)\in\mathcal{A}^{\mathcal{F}}} c^f_{nm}(y^b_{nm} + y^d_{nm}) \quad + \sum_{(nm)\in\mathcal{A}^{\mathcal{G}}} c^s_{nm} y^s_{nm} + \sum_{(nm)\in\mathcal{A}^{\mathcal{O}}: i(n)\notin I^H} c^o_{nm} y^o_{nm} \tag{5.1a}$$

$$\text{s.t.} \sum_{l:(lm)\in\mathcal{A}^{\mathcal{O}}} y^o_{lm} + \sum_{l:(lm)\in\mathcal{A}^{\mathcal{G}}} y^s_{lm} = \sum_{n:(mn)\in\mathcal{A}^{\mathcal{F}}} (y^b_{mn} + y^d_{mn}) + \sum_{n:(mn)\in\mathcal{A}^{\mathcal{G}}} y^s_{mn} \;\forall\, m \in \mathcal{N} \tag{5.1b}$$

$$\sum_{l:(lm)\in\mathcal{A}^{\mathcal{F}}} (y^b_{lm} + y^d_{lm}) + \sum_{l:(lm)\in\mathcal{A}^{\mathcal{G}}} y^s_{lm} = \sum_{n:(mn)\in\mathcal{A}^{\mathcal{G}}} y^s_{mn} + \sum_{n:(mn)\in\mathcal{A}^{\mathcal{O}}} y^o_{mn} \qquad \forall\, m \in \mathcal{N} \tag{5.1c}$$

$$\sum_{(lm):\tau(l)\le t, \tau(m)>t} (y^b_{lm} + y^d_{lm} + y^s_{lm}) \le \sum_{(mn):\tau(m)-t^{dp}\le t<\tau(m)} y^o_{mn} \qquad \forall\, t \tag{5.1d}$$

$$\sum_{l:(lm)\in\mathcal{A}^{\mathcal{O}}} y^o_{lm} = \sum_{n:(mn)\in\mathcal{A}^{\mathcal{D}}} \delta_{mn} \qquad \forall\, m \in \mathcal{N} \tag{5.1e}$$

$$\sum_{l:(lm)\in\mathcal{A}^{\mathcal{D}}} \delta_{lm} = \sum_{n:(mn)\in\mathcal{A}^{\mathcal{O}}} y^o_{mn} \qquad \forall\, m \in \mathcal{N} \tag{5.1f}$$

$$y^b_{nm} = x_{nm} \qquad \forall\, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{5.1g}$$

$$y^d_{nm} \le H x_{nm} \qquad \forall\, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{5.1h}$$

$$\sum_{(mn)\in\mathcal{A}^{\mathcal{D}}(0)} \delta_{mn} + \sum_{(mn)\in\mathcal{A}^{\mathcal{O}}(0)} y^o_{mn} \le C \tag{5.1i}$$

$$\delta_{nm} \ge 0 \quad \forall\, (nm) \in \mathcal{A}^{\mathcal{D}} \qquad y^b_{nm}, y^d_{nm} \ge 0 \qquad \forall\, (nm) \in \mathcal{A}^{\mathcal{F}}$$

$$y^s_{nm} \ge 0 \quad \forall\, (nm) \in \mathcal{A}^{\mathcal{G}} \qquad y^o_{nm} \ge 0 \qquad \forall\, (nm) \in \mathcal{A}^{\mathcal{O}} \tag{5.1j}$$

The objective function (5.1a) sums the costs for block and sit time, as well as overnights away from crew bases.

Equations (5.1b) and (5.1c) control flow for each duty period, according to the following logic. Overnight arcs and sit arcs must be followed by block time, a deadhead arc, or a sit arc. Flight arcs (block time or a deadhead arc) lead to a sit arc or an overnight arc.

To ensure that duty periods are followed by overnights, (5.1d) require that for any time t, the total duty flow is no more than the total overnight flow starting between time $t+1$ and a time t^{dp} later. For example, if 13 flights are airborne, no crews are deadheading, and 5 crews are on duty and on the ground at 8:00am, then at least 18 crew rest periods must start between now and 12 hours later at 8:00pm.

Equations (5.1e) and (5.1f) conserve flow between duty period arcs and overnight arcs, so that overnight flow is followed by equal duty period flow, and duty period flow is followed by equal overnight flow.

Equations (5.1g) require that each flight from the MSP solution will be flown by a crew. In (5.1h), deadheads are constrained to the available flights, but multiple crews may deadhead on the same flight.

Constraint (5.1i) restricts the duty flow counted at $t = 0$ to the number of available crews in a day.

The remainder of the constraints require nonnegativity of all variables.

5.2 Validation of the SCSP

As discussed by Andersson et al. (1998), exact crew pairing models are very large and complex, even for small problems. A string model similar to the one developed for validation of the maintenance surrogate model can be constructed to more precisely estimate the optimal crew costs associated with a given schedule. To confirm that the SCSP provides a reasonable relative approximation of crew costs, we compare the objective values obtained by SCSP for the test cases to those obtained from a crew string model.

The crew string model includes a pairing $p \in P$ for each individual crewmember itinerary. The string model also enforces the FAA block time limit of 8 hours per day, denoted by t^b and the cycle time is c. Also, δ^0_{np} is 1 if the start of duty period $p \in P$ occurs at node $n \in \mathcal{N}$ and 0 otherwise, and δ^1_{np} marks the end of pairing p similarly. Given these additional pieces of notation and those defined for the SCSP above and assuming identical cost coefficients for each crewmember, the crew string model is as follows.

$$\min \sum_{p \in P} \{ \sum_{(nm) \in \mathcal{A}^{\mathcal{F}}} c^f_{nm}(y^b_{nmp} + y^d_{nmp}) \quad + \sum_{(nm) \in \mathcal{A}^{\mathcal{G}}} c^s_{nm} y^s_{nmp} \} + \sum_{(nm) \in \mathcal{A}^{\mathcal{O}} : i(n) \notin I^H} c^o_{nm} y^o_{nm} \tag{5.2a}$$

$$\text{s.t. } \delta^0_{mp} + \sum_{l:(lm) \in \mathcal{A}^{\mathcal{G}}} y^s_{lmp} = \sum_{n:(mn) \in \mathcal{A}^{\mathcal{F}}} (y^b_{mnp} + y^d_{mnp}) + \sum_{n:(mn) \in \mathcal{A}^{\mathcal{G}}} y^s_{mnp} \qquad \forall\, m \in \mathcal{N}, p \in P \tag{5.2b}$$

$$\sum_{l:(lm) \in \mathcal{A}^{\mathcal{F}}} (y^b_{lmp} + y^d_{lmp}) + \sum_{l:(lm) \in \mathcal{A}^{\mathcal{G}}} y^s_{lmp} = \sum_{n:(mn) \in \mathcal{A}^{\mathcal{G}}} y^s_{mnp} + \delta^1_{mp} \qquad \forall\, m \in \mathcal{N}, p \in P \tag{5.2c}$$

$$\sum_{p \in P} \delta^1_{mp} = \sum_{n:(mn) \in \mathcal{A}^{\mathcal{O}}} y^o_{mn} \qquad \forall\, m \in \mathcal{N} \tag{5.2d}$$

$$\sum_{p \in P} \delta^0_{mp} = \sum_{l:(lm) \in \mathcal{A}^{\mathcal{O}}} y^o_{lm} \qquad \forall\, m \in \mathcal{N} \tag{5.2e}$$

$$\sum_{n \in \mathcal{N}} \delta^0_{np} \leq 1 \qquad \forall\, p \in P \tag{5.2f}$$

$$\sum_{n \in \mathcal{N}} \delta^1_{np} \leq 1 \qquad \forall\, p \in P \tag{5.2g}$$

$$\sum_{p \in P} y^b_{nmp} = x_{nm} \qquad \forall\, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{5.2h}$$

$$\sum_{p \in P} y^d_{nmp} \leq H x_{nm} \qquad \forall\, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{5.2i}$$

$$\sum_{(lm) \in \mathcal{A}^{\mathcal{F}}} t_{i(l)i(m)} y^b_{lmp} \leq t^b \qquad \forall\, p \in P \tag{5.2j}$$

$$\sum_n t_{i(n)}\delta^1_{np} - \sum_n t_{i(n)}\delta^0_{np} \le t^{dp} \qquad \forall p \in P \tag{5.2k}$$

$$\sum_n t_{i(n)}\delta^1_{np} - \sum_n t_{i(n)}\delta^0_{np} \ge 0 \qquad \forall p \in P \tag{5.2l}$$

$$\sum_{(mn)\in\mathcal{A}(0)} \{\sum_{p\in P}(y^b_{mnp} + y^d_{mnp} + y^s_{lmp}) + y^o_{mn}\} \le C \tag{5.2m}$$

$$\begin{aligned}
y^s_{nmp} &\in \{0,1\} && \forall\,(nm) \in \mathcal{A}^{\mathcal{G}}, p \in P \\
y^b_{nmp}, y^d_{nmp} &\in \{0,1\} && \forall\,(nm) \in \mathcal{A}^{\mathcal{F}}, p \in P \\
y^o_{nm} &\in Z^+ && \forall\,(nm) \in \mathcal{A}^{\mathcal{O}} \\
\delta^0_{np}, \delta^1_{np} &\in \{0,1\} && \forall\,(n) \in \mathcal{N}, p \in P
\end{aligned} \tag{5.2n}$$

The constraints are similar to those of the SCSP, except for the redefinition of the δ variables described above, and the addition of (5.2j) and (5.2k), to enforce FAA block time and duty period rules, respectively. Also, equations (5.2d) and (5.2e) require that each pairing $p \in P$ end with an overnight, and that overnights flow into pairings. Constraints (5.2f) and (5.2g) limit each pairing to a single start and finish, so that a pairing can only include one duty period, and (5.2l) requires that pairings start before they end. For an exact solution, the flow variables within a duty period and the start and end markers must be binary, and the overnight flow must be integer.

The set of nine test schedules was evaluated in both the SCSP and the crew string model, and the relative objective values were compared. Figure 5.2 shows the resulting relationship between objective values for the two models. The R^2 value for these nine test schedules is 0.85, confirming a reasonably strong linear relationship. To truly validate the surrogate model, larger, more realistic schedules should be input to both models, and the solutions should be similarly compared. For our test cases, these results provide assurance that the surrogate is reasonably useful for representation of crew costs in the MSP.

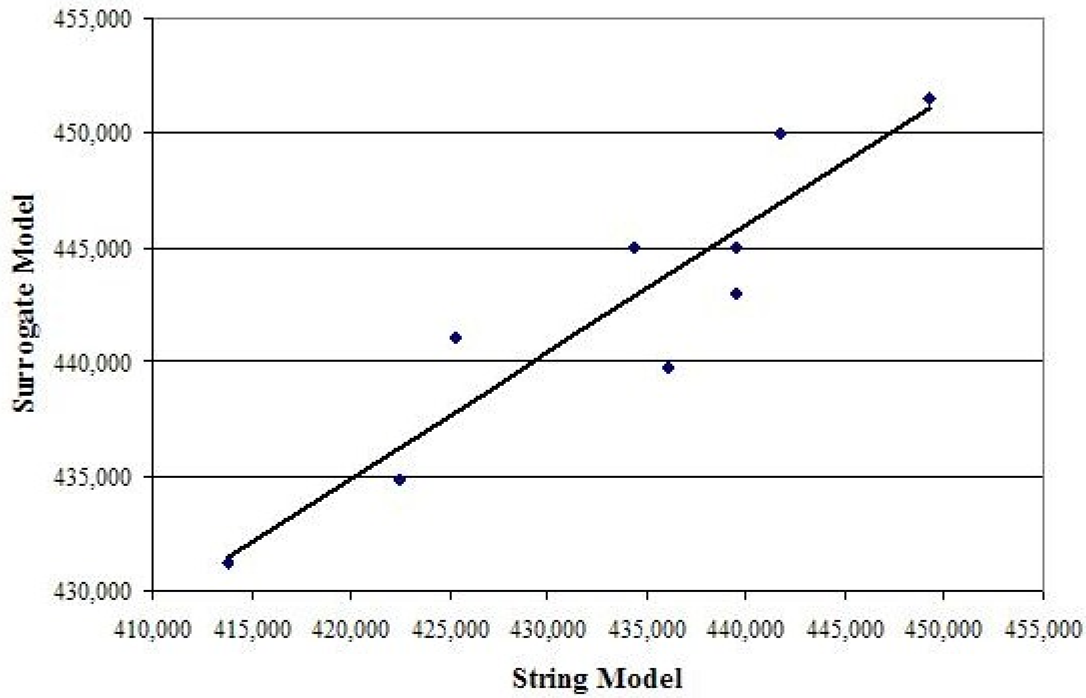

Figure 5.2: Objective values of surrogate crew model vs. string model.

5.3 Random elements of the SCSP

Several data elements of the SCSP are not fixed. To construct a more accurate representation of crew costs, we may model the sources of randomness for inclusion in our analysis. The total crew costs for the SCSP with randomness may well be higher, and thus more conservative, than that of the SCSP without randomness. The inclusion of random behavior can shape the master schedule to be more robust to imperfect weather and other factors that impact crew costs.

For example, the number of crews available to fly at any time may vary from day to day, due to training requirements, illness, and other factors. To model randomness in the number of crews, represented as C in SCSP, we may replace C with the random variable $\hat{C}$. An airline can model the disribution of $\hat{C}$ by analyzing historical crew availability rates.

Crew pay rates also vary with crew assignment. Crew cost coefficients depend on the actual pilots assigned to the pairing. A senior captain makes significantly more money than a junior first officer. Pilots in the United States generally bid on trips and are assigned in order of seniority, and each pilot has unique preferences for trip timing and destinations. To reflect the fact that lower-paid junior crewmembers are more likely to have to fly to less desirable places, we may develop a distribution that scales the cost coefficients so that some cities are more likely to have higher crew costs than others.

Unplanned schedule deviations also cause variability in crew costs. Frequently, the actual trip flown by a pilot differs from the planned trip. For example, a weather delay in one city may make the pilots miss their plane change in the next city, and they may be assigned to a different connecting flight. They may also deadhead halfway through a disrupted trip to get back on schedule. While managing the actual crew reassignments falls within the set of recovery problems, modelling schedule deviations early in the planning process may improve schedule robustness and provide a more accurate estimation of the relative crew cost of a schedule.

The following discussions of the random aspects of the crew scheduling problem outline approaches for modelling this random behavior. These models can be used with appropriate data and L-shaped method solution techniques to improve the utility of the SCSP.

5.3.1 Random crew costs

Each airline has a distribution of the seniority of its pilots and first officers. Many seniority levels exist, as few pilots have spent the same amount of time with their airline. To approximate pilot seniority, we will define a limited number of aggregate levels, such as 0-3 years or 16-20 years with the airline. The probability of randomly choosing a pilot from a certain bracket can be calculated according to the actual seniority distribution of an airline's pilots. Rather than calculating actual pay rates, we use a baseline pay rate and a cost multiplier for each position (captain or first officer) and seniority level. A crewmember's hourly rate is therefore calculated as the base rate times the crewmember's multiplier. A simple example of a pay distribution for a single fleet type appears in Table 5.1.

Position	**Seniority Level**	**Multiplier**	**Probability**
Captain	1	2.0	0.20
	2	1.8	0.25
	3	1.7	0.18
	4	1.5	0.22
	5	1.4	0.15
First Officer	1	1.2	0.10
	2	1.1	0.35
	3	1.0	0.25
	4	0.7	0.20
	5	0.4	0.10

Table 5.1: Sample crew pay multiplier distribution.

To solve the SCSP with randomness in crew pay, we may construct a set of crew assignment scenarios, solve SCSP with each scenario, and calculate the expected crew costs given these solutions. For each crew assignment scenario s, the assigned crew members are each paid at one of the levels shown in the table. Let μ_s^1 be the cost multiplier for the captain in scenario s on leg (nm), and let μ_s^2 represent the multiplier for the first officer. To maintain simplicity, crew pay levels are randomly drawn for each leg, independent of all other legs. In other words, we will not attempt to conserve the flow (or pairing assignment) of individual pilots working at a specific pay level. The SCSP objective function for a given scenario s may be stated as follows.

$$\min \sum_{(nm)\in\mathcal{A}} (\mu_s^1 + \mu_s^2)[(c_{nm}^f y_{nm}^b + c_{nm}^s y_{nm}^d + c_{nm}^f y_{nm}^s s) + \sum_{(nm):i(n)\notin I^H} c_{nm}^o y_{nm}^o] \qquad (5.3)$$

If the scenario set is not large, we may construct all possible combinations of crew pay levels and solve SCSP for each, before calculating the overall expected crew cost value. If too many scenarios exist to solve SCSP for all of them within the available computation time, we can sample them and solve SCSP for the selected subset. To sample scenarios, we simply draw from the crew pay distribution for each pairing and insert the pay rate outcomes into the objective function. For example, with probability 0.22, we will draw a captain with seniority level 4 to fly a given flight leg, and with probability 0.25, we will draw a first officer with seniority level 3. The first officer is paid at the baseline rate for that duty since his or her multiplier is 1.0, and the captain is paid at 1.5 times the baseline rate. For simplicity, we sample with replacement.

If an airline has a larger percentage of captains than first officers for a certain fleet, we may include a Bernoulli random variable that indicates whether the second crewmember is a captain or a first officer. If the Bernoulli indicates that a captain will occupy the second position, the captain pay distribution applies to both crewmembers. Let $\alpha = 1$ if a captain will fill the second position. The corresponding SCSP objective function is now

$$\min \sum_{(nm)\in\mathcal{A}} ((1+\alpha)\mu_s^1 + (1-\alpha)\mu_s^2)[(c_{nm}^f y_{nm}^b + c_{nm}^f y_{nm}^d s + c_{nm}^s y_{nm}^s) + \sum_{(nm):i(n)\notin I^H} c_{nm}^o y_{nm}^o] \tag{5.4}$$

If we account for the possibility of two captains flying together, we must draw from the Bernoulli distribution as well as the seniority distribution when generating the scenarios. In any case, after solving SCSP for the chosen scenarios, we can calculate the expected SCSP objective value and use the results to continue solution of the MSP.

The constraints remain as presented previously in (5.1a)-(5.1j).

5.3.2 Random schedule deviations

Schedule deviations may be reflected by adjusting the flight times associated with the MSP solution. We may move the start and end times of any input x_{nm} according to the location and time of year, to reflect frequent weather problems in certain locations, holding patterns, unplanned maintenance, and other external forces. The probability distribution for flight time adjustments may also depend on time of day, so that larger adjustments occur later in the day when the "domino effect" of earlier delays has impacted the schedule.

In the distribution shown in Table 5.2, airports are categorized according to schedule disruption vulnerability due to weather, where category A cities are the most vulnerable (such as Minneapolis in the wintertime) and category C cities are the least likely to suffer weather delays. The day is divided as shown, and the numbers in the table display the probability of a disruption as well as the discretely distributed disruption length, in number of time periods. For example, a category B city has a 20% chance of experiencing a delay in the evening, and the delay length is 2 time periods with probability 0.8.

Vulnerability	**Morning**		**Afternoon**		**Evening**	
	p_{delay}	Length (p)	p_{delay}	Length (p)	p_{delay}	Length (p)
A	0.1	1 (0.8)	0.2	2 (0.75)	0.35	3 (0.2)
		2 (0.18)		3 (0.2)		4 (0.6)
		3 (0.02)		4 (0.05)		5 (0.2)
B	0.05	1 (.7)	0.1	2 (.65)	0.2	2 (0.8)
		2 (.3)		3 (.3)		3 (.13)
				4 (.05)		4 (.02)
C	0.02	1 (.9)	0.07	1 (.5)	0.15	2 (.4)
		2 (.05)		2 (.4)		3 (.4)
		3 (.05)		3 (.1)		4 (.1)
						5 (.1)

Table 5.2: Sample schedule disruption distribution.

To increase the realism of the schedule disruption model, we may create a similar distribution for arrival delays, such as air traffic control-ordered holding patterns, that will extend the duration of the flight arc. Such a distribution may also allow flights to be slightly shorter than planned, due to tailwinds or crew efforts to regain lost time after a delay.

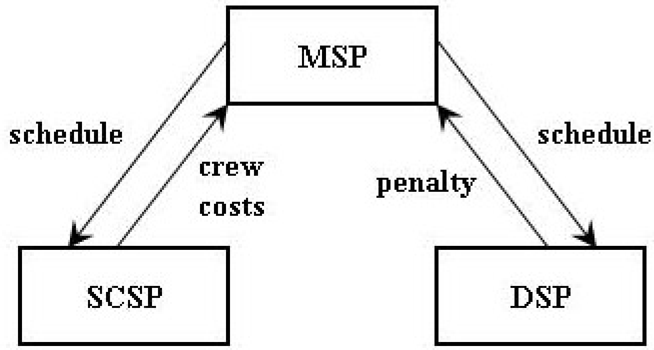

Figure 5.3: Structure of MSP with crew scheduling and schedule disruptions.

To model schedule disruptions, we insert a new subproblem for the MSP, as shown in Figure 5.3. This subproblem transforms the x_{lm} to $\hat{x}^0_{np}$ via random disruptions $x^*_{lm/np}$, where (np) is an arc close in time to $(lm) \in \mathcal{A}^{\mathcal{F}}$. For example, suppose that we have three airports in the network, and $x_{1,11}$ is a flight from airport 1 to airport 2 that requires 3 time periods for flight, so that it starts at node 1 and ends at node 11. If the random disruption generator delays the flight's takeoff by one time period, it becomes $x_{4,14}$, which starts at node 4 and ends at node 14. This delay is illustrated in Figure 5.4, where the solid arrows represent scheduled flights, and the dashed line depicts the delayed flight. We represent this disruption as $x_{1,11/4,14}$. After the schedule has been disrupted, the associated objective value, multiplied by a penalty, is added to the MSP objective value.

Since flight delays will precipitate through the network, the transformation function is nontrivial. In real-world airline operations, airlines aim to minimize changes to the overall schedule when a disruption occurs, as described by Kohl et al. (2004). This minimization responds to the assumption that the initial schedule is optimal, and by limiting the deviations from the schedule caused by disruption, movement away from optimality is minimized. Therefore, the model for the disrupted schedule minimizes the sum of changes in flow on all arcs when disruptions are randomly generated and fixed in the new schedule.

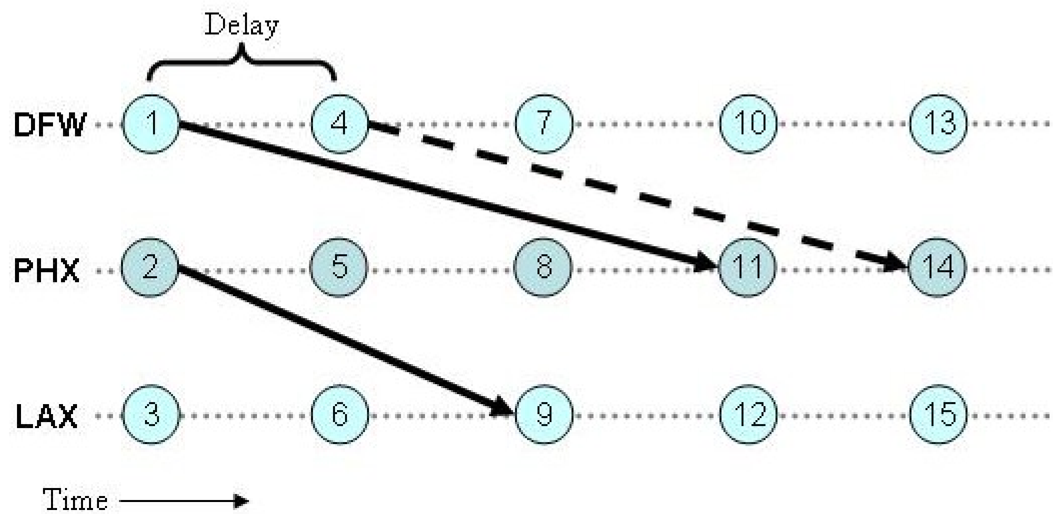

Figure 5.4: Flight delay of one time period.

The model for the disrupted schedule problem (DSP) penalizes changes in the schedule flow on any arc. Decision variables $\hat{x}_{nm}$ represent flight flow from node n to node m in the disrupted schedule. Increases and decreases between the original and disrupted flow on each arc $(nm) \in \mathcal{A}$ are denoted by Δ^+_{nm} and Δ^-_{nm}, respectively. Given these definitions and those from the MSP and SCSP above, the model is as follows.

Sets:

$\mathcal{A}^\Delta$ Set of arcs $(nm) \in \mathcal{A}^\mathcal{F}$ disrupted by random disruption generator

Data:

P Penalty for change in flow on an arc (nm)

Input from random schedule disruptor:

$x^*_{lm/np}$ Arc corresponding to $(lm) \in \mathcal{A}^\mathcal{F}$, disrupted to $(np) \in \mathcal{A}^\mathcal{F}$

Decision variables:

$\hat{x}_{nm}$ Flow from n to m after schedule disruption, $(nm) \in \mathcal{A}$

Δ^{+}_{nm} Positive change in flow on $(nm) \in \mathcal{A}$

Δ^{-}_{nm} Negative change in flow on $(nm) \in \mathcal{A}$

DSP:

$$\min \sum_{(nm)\in\mathcal{A}} P(\Delta^{+}_{nm} + \Delta^{-}_{nm}) \tag{5.5a}$$

$$\text{s.t.} \quad x_{nm} - \hat{x}_{nm} - (\Delta^{+}_{nm} - \Delta^{-}_{nm}) = 0 \qquad \forall\,(nm) \in \mathcal{A} \tag{5.5b}$$

$$\sum_{lm\in\mathcal{A}^{\Delta}} x^{*}_{lm/np} \le \hat{x}_{np} \qquad \forall\,(np) \in \mathcal{A}^{\Delta} \tag{5.5c}$$

$$\sum_{m} \hat{x}_{nm} \le r^{-}_{i(n)} \qquad \forall\, n \in \mathcal{N}, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{5.5d}$$

$$\sum_{n} \hat{x}_{nm} \le r^{+}_{i(m)} \qquad \forall\, m \in \mathcal{N}, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{5.5e}$$

$$\sum_{n} \hat{x}_{nm} \le g_{i(m)} \qquad \forall\, t, m \in \mathcal{N}(t), (nm) \in \mathcal{A}^{\mathcal{G}} \tag{5.5f}$$

$$\sum_{(nm)\in\mathcal{A}(0)} \hat{x}_{nm} \le V \tag{5.5g}$$

$$\sum_{(mn)\in\mathcal{A}^{\mathcal{F}},\tau(p)-\gamma(i(n))<\tau(n)\le\tau(p),i(n)=i(p)} \hat{x}_{mn} \le \hat{x}_{pq} \qquad \forall\,(pq) \in \mathcal{A}^{\mathcal{G}} \tag{5.5h}$$

$$\sum_{m} \hat{x}_{nm} - \sum_{m} \hat{x}_{mn} = 0 \qquad \forall\, n \in \mathcal{N} \tag{5.5i}$$

$$\hat{x}_{nm} \ge 0 \qquad \forall\,(nm) \in \mathcal{A} \tag{5.5j}$$

The objective of DSP is to minimize deviations from the original schedule while maintaining feasibility. The constraints are nearly the same as for MSP. Equations (5.5b) record the increases and decreases in flow on each arc. In equations (5.5c), the flows on the disrupted arcs are required to be at least as great as prescribed by the disruptor, so that the rest of the schedule must adjust for them. Constraints (5.5d) through (5.5j) simply maintain schedule feasibility as prescribed by MSP, except that wrap constraints (3.1h) are omitted since disruptions generally vary from day to day.

By randomly generating numerous disruption scenarios and solving DSP with $\hat{x}$ from each, we can estimate the effect of disruption on a schedule. A more robust schedule is easier to recover with minimum adjustment, and has a lower DSP objective value. This will shape the MSP towards a schedule that is less sensitive to random disruptions. However, we must adjust the L-shaped method to account for the fact that each disruption scenario $\hat{x}$ is different from x.

5.3.3 Cut generation with schedule disruptions

The L-shaped method requires that cuts be generated with respect to the current master problem solution x. Therefore, the transformation of x to $\hat{x}$ requires special treatment to allow proper execution of the algorithm. We therefore modify subproblem constraints (A.9e) to account for the transformation, so that we have $Wy = r - T\zeta x$, where ζ is a binary matrix that reflects the transformation of x to $\hat{x}$. If we let $\hat{T} = T\zeta$, we effectively account for the schedule disruption by introducing randomness into T. With s as the number of disruption scenarios, we can generate optimality cuts as as $\eta \geq \sum_s p_s \pi_s (r - \hat{T}_s x)$, and feasibility cuts as $\sum_s p_s \mu_s (r - \hat{T}_s x) \leq 0$.

5.4 Implementation of the crew cost approximation models

To demonstrate the models defined above, we solved the models using the L-shaped method. We first tested the models with the same 5-airport networks as we used for the maintenance experiment, and then employed models with up to 30 airports to demonstrate scalability. The sizes of the test problems are listed in Appendix B. A detailed explanation of the methods used to construct schedule disruption distributions may be found in Appendix C. We omitted randomness in crew pay for simplicity, so that the objective function coefficients in SCSP are constant.

5.4.1 Computations and results

For each test problem, we solved MSP with the SCSP subproblem first with Benders' decomposition, and then using the L-shaped method with the disruption penalty from DSP. For consistency, we calculated 100 possible disruptions per iteration to establish bounds on the objective value, and to establish a master schedule solution. Each test case solution was considered optimal when the gap between the upper and lower bound on the MSP objective function fell below 0.1% of the lower bound value. To refine the objective value for the cases that included DSP, we solved DSP iteratively with this master schedule solution and new randomly generated schedule disruptions, calculating a confidence interval on the objective value at each iteration as described in Section 3.4.

Given each test problem, we employed the measures described in Section 3.4 to compare one day's flight schedule from both the mean value problem (SCSP only, without randomness) and the schedule with the disruption penalty. Results of the crew schedule experiment appear in Table 5.3, where "Undis" denotes the undisrupted schedule, "Dis" indicates that the disruptions are considered, and LCL and UCL are lower confidence level and upper confidence level, respectively. For the undisrupted cases, we first solved the MSP and SCSP for the test networks, without DSP. We then conducted the confidence interval procedure described in Section 3.4 using the resulting schedule and DSP. The resulting confidence intervals appear in the "Undis:LCL" and "Undis:UCL" columns. We then repeated the procedure, but began by solving MSP with both SCSP and DSP, and the results appear in the "Dis" columns. The values indicate profit generated by each schedule, calculated as the estimated schedule revenue less the approximated crew costs and disruption penalty. Note that the coefficients included in these test cases were rescaled from those used in other chapters, for demonstration purposes.

Table 5.3 shows that the expected profit was greater when disruptions were considered, even though the disruption subproblem had little weight in the objective

		Objective value			
Net	**Cities**	**Undis:LCL**	**Undis:UCL**	**Dis:LCL**	**Dis:UCL**
1	5	2,357,366	2,357,383	2,381,295	2,381,314
2	5	1,954,053	1,954,077	2,033,189	2,033,517
3	5	2,376,967	2,376,983	2,441,481	2,441,505
4	5	2,624,525	2,624,544	2,666,519	2,666,534
5	5	2,031,867	2,031,953	2,164,099	2,164,126

Table 5.3: Crew schedule experiment results.

function. The cuts generated by the disruption subproblem apparently shaped the schedule towards higher expected profits. A comparison of schedules with and without consideration of the disruption should illuminate these changes further.

As described in Section 3.4, Table 5.4 contains the following metrics for comparing two schedules:

- The number of flights that appear in both schedules
- The number of flights that are nearly the same in both schedules, defined as flights that share an O-D pair but fly two or fewer time increments later in the second schedule
- The total number of flights in each schedule
- The number of "extended" ground arcs (EGAs), or consecutive ground arcs that exceed the minimum turn time in each schedule, as a measure of schedule robustness

We also include the number of EGAs scheduled later in the day, when recovery is generally more prevalent to reduce the propagation from disruptions caused earlier in the day.

For each test case, the most profitable schedule for the undisrupted crew schedule (Undis) is compared with that with disruptions (Dis). These test cases involved only a single fleet.

Test Case	Identical flights	"Near" flights	Total Flights Undis	Total Flights Dis	EGAs Undis	EGAs Dis	Night EGAs Undis	Night EGAs Dis
1	1	5	55	53	98	100	23	32
2	7	10	50	58	62	98	19	46
3	9	3	48	46	130	92	25	28
4	7	41	64	59	94	80	24	28
5	3	1	49	59	104	96	30	34

Table 5.4: Flight count summary.

Table 5.4 shows that the number of flights changed when randomness was considered, and the flight times changed considerably in most cases, except for test case 4. Therefore, the consideration of disruption seems to shape the schedule, but not following any consistent pattern. Additionally, the schedules with consideration of disruption did not necessarily contain more EGAs. If we examine the timing of the EGAs, we see that those scheduled for later times in the day increase, reflecting the greater need for EGAs to reduce the effect of delay propagation. The disruption subproblem has shifted the EGAs to facilitate recovery from disruptions.

5.5 Limitations and extensions

As a surrogate model, SCSP by definition is inexact. For example, it does not enforce FAA maximum block time rules. To enforce block time rules, we would have to increase the model dimensionality to include an immense number of possible crew pairings. An algorithm that employs a fast, more accurate crew cost model and advanced solution techniques could replace SCSP, thereby improving the utiliy of MSP. The tradeoff between accuracy and speed must be considered within the context of the end-user.

An airline may want to investigate the change in crew costs due to schedule disruptions. To perform such an analysis, we can compare the crew costs from SCSP with undisrupted schedule x from the MSP to those from schedule with disruptions $\hat{x}$ from DSP. A confidence interval for $f(x) - f(\hat{x})$ provides an estimate of the impact of schedule disruptions on crew cost, assuming that the disruption distributions are modelled accurately.

While the data used in our experiments were artificial, actual data would be relatively simple to collect for a real flight network. Each airport has historical weather records, and the probability of disruption can be statistically modelled. The time windows should be discretized carefully, since each region experiences different traffic and weather patterns at different times of day. Further, distributions should be modelled seasonally, since disruption factors can vary considerably at different times of the year.

CHAPTER 6

Revenue Approximation Under Uncertainty

In Chapter 3, we introduced the MSP with the cost contribution from the subproblems $b \in \mathcal{B}$ represented as $\sum_{b \in \mathcal{B}} f_b(x)$. Each flight generates revenue while incurring fuel and crew costs, which are discussed in previous chapters. Revenue per flight depends on the number of passengers carried and the price each paid for a ticket, where the price depends on the passenger's origin and destination (O-D) and other factors. The revenue generated by a flight is a random variable, and warrants consideration as such.

Nearly every airline in the world sells seats on every flight at a range of prices and classes of service. First and coach cabin seat capacity on each airplane is fixed, although Pak et al. (2003) discuss the potentially profitable but probably impractical concept of class-convertible seating. Within each fixed class, seats are sold at different prices and at different times. In order to plan the most profitable and robust schedule, a subproblem of the model described by equations (3.1a)-(3.1i) should account for this random behavior. The following discussion explores the application of stochastic programming in airline schedule planning to find the most profitable schedule when the demand for capacity is random. The model presented will lead to a schedule that allocates capacity, or seats, to maximize expected revenue. We will begin with an expansion of the MSP that includes fleet assignments, followed by a model for assigning capacity based on random demand and revenue.

6.1 MSP with fleeting (MSPF)

Potential revenue on any flight depends on seat supply and demand, as well as the ticket price per seat. Airlines estimate demand using historical data and market analyses. These analyses account for such events as the relocation of a large firm that requires extensive business travel to a new city, or the decline of tourism in an area following a major natural disaster. Competition from other airlines may also influence demand. The number of available seats depends on the size of the aircraft assigned to the flight, where a larger plane offers more seats than a smaller one. Flights are assigned to fleets based on projected demand as well as aircraft availability.

To model fleet assignment, we may simply add a fleet index to the decision variables x_{nm} in the MSP. We may now treat MSP as a multicommodity flow model, and must increase the number of flow conservation and plane count constraints so that each fleet maintains a feasible flow with the available aircraft. The set and data definitions are the same as for the original MSP, except as noted. Also, we will assume that each gate at an airport can accommodate any type of aircraft, so that ground capacities of airports are not fleet-specific.

Sets:

K Fleets

Data:

t_{ijk} Travel time from location i to location j with fleet k

c_{ijk} Fuel cost for travel from i to j with fleet k(zero for ground arcs)

V_k Number of airplanes of fleet type k available in the system

Decision variables:

x_{nmk} Flow from node n to node m using fleet k, $(nm) \in \mathcal{A}, k \in K$

MSPF:

$$\min \sum_{(nm)\in\mathcal{A}^{\mathcal{F}}} \sum_{k} c_{i(n)i(m)k} x_{nmk} + \sum_{b\in\mathcal{B}} f_b(x) \tag{6.1a}$$

$$\text{s.t.} \quad \sum_{k}\sum_{m} x_{nmk} \leq r^{-}_{i(n)} \quad \forall\, n \in \mathcal{N}, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{6.1b}$$

$$\sum_{k}\sum_{n} x_{nmk} \leq r^{+}_{i(m)} \quad \forall\, m \in \mathcal{N}, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{6.1c}$$

$$\sum_{k}\sum_{n} x_{nmk} \leq g_{i(m)} \quad \forall\, t, m \in \mathcal{N}(t), (nm) \in \mathcal{A}^{\mathcal{G}} \tag{6.1d}$$

$$\sum_{(nm)\in\mathcal{A}(0)} x_{nmk} \leq V_k \quad \forall\, k \in K \tag{6.1e}$$

$$\sum_{(mn)\in\mathcal{A}^{\mathcal{F}}, \tau(p)-\gamma(i(n))<\tau(n)\leq\tau(p)} x_{mnk} \leq x_{pqk} \quad \forall\, (pq) \in \mathcal{A}^{\mathcal{G}}, k \in K \tag{6.1f}$$

$$\sum_{m} x_{nmk} - \sum_{m} x_{mnk} = 0 \quad \forall\, n \in \mathcal{N}, k \in K \tag{6.1g}$$

$$x_{nmk} - x_{n'm'k} = 0 \quad \forall\, k \in K, (nm) \in \mathcal{A}, i(n) = i(n'), i(m) = i(m'), \tau(n') = \tau(n) + c \tag{6.1h}$$

$$x_{nmk} \in Z^{+} \quad \forall\, (nm) \in \mathcal{A}, k \in K \tag{6.1i}$$

The solution to this model will be similar to the MSP solution, except that MSPF includes fleet assignment information in the schedule. The revenue approximation subproblem can be used to estimate revenue for the resulting schedule.

6.2 Capacity allocation model (CAM)

Maximizing revenue requires optimal allocation of seats to classes, as well as fleets to flights. While assigning capacity is relatively simple, revenue modelling can be complex. Different types of passengers buy different classes of tickets at different times, depending on the availability of their desired class of service. Modelling demand requires clever partitioning of both time and passenger types. An airline can estimate distributions for demand using historical data of actual passengers by class, as well as forecasts by class. Note that the itineraries used to route passengers from an origin to a destination encapsulate the time of the day and week, so that different distribution parameters at different travel times may be reflected.

The demand data must be modelled carefully. Passengers usually want to fly on a certain day within a time window, such as early morning. Therefore, several potential itineraries connecting the same two cities and starting around the same time may be pooled to satisfy demand for a time window. These itineraries may connect through different cities. To account for this, we introduce the set W, where $w \in W$ corresponds to a time window and an O-D pair. We will refer to a time window and O-D pair combination as a *basket*. For example, $w =$ (morning, Tucson, Baltimore) may include these itineraries: 9am departure, Tucson-Cincinnati-Baltimore; 7am departure, Tucson-Atlanta-Baltimore; 6:30am departure, Tucson-Salt Lake City-Baltimore. All three itineraries leave Tucson in the morning and end in Baltimore, but the departure times and connection cities are different. The set W includes all baskets, i.e., all combinations of time windows (such as morning, noon, afternoon, evening) and origin and destination cities. In the implementation that follows, each basket w will contain no more than three itineraries.

Each itinerary $i \in \mathcal{I}$ consists of legs $(nm) \in \mathcal{A}$, and each flight services classes $c \in C$. More than one class may occupy a cabin, such as deep-discount and full fare coach passengers. Each fleet has a different seat capacity, represented by S_k. The demand for seats in each class and basket varies with the time of day, season, departure and arrival cities, fares offered, and other factors. Further, ticket demand for different classes varies with the amount of time until takeoff; for example, demand for discount tickets is higher well in advance of the flight, while more business-fare tickets are requested within one week of departure. Airline marketing researchers attempt to predict ticket demand, and bid-pricing algorithms define methods for maximizing revenue by changing seat availability in each class as tickets are sold. To represent this phenomenon in a much simpler way, we denote the ticket purchases, or realized demand, for a class c in basket w as d_{wc}. These data are effectively a compression of the ticket selling process, and indicate the final ticket sales mix for the flights in a basket. An airline can construct this prediction using historical data and marketing research figures. The average fare per class on each itinerary can be estimated similarly, since the actual sale prices of the tickets in a fare class may vary, depending on the ticket pricing system used by the airline. We represent the average fare for class c on itinerary i as v_{ic}.

Given the flights in the master schedule assigned to fleet k, we allocate seat capacity by itinerary with variables y^r_{ic} according to final ticket sales d_{wc}. The superscript r simply designates these variables as attributes of the revenue surrogate model, and distinguishes the y^r_{ic} variables from other y variables in other subproblems, such as the y^b_{nm} block time flow variables in SCSP. Initially, we will treat d_{wc} as fixed demand for each class and basket.

The model that follows resembles that presented by Clarke and Gong (1995) for capacitating communications networks with O-D demand. For simplicity, we assume no overbookings, no-shows, cancellations, or group bookings, as defined in Chapter 2.

Sets:

$\mathcal{A}(t)$	Arcs that are active at time t, i.e., $\tau(n) \leq t$ and $\tau(m) > t$ for $(nm) \in \mathcal{A}(t)$
W	Baskets, or time window/O-D pair combinations
$\mathcal{I}$	Itineraries, or travel routes between O-D pairs
C	Fare classes

Data:

v_{ic}	Average fare for class c on itinerary i
σ_{nmi}	1 if leg $(nm) \in \mathcal{A}$ is in itinerary $i \in \mathcal{I}$, else 0

η_{iw} 1 if itinerary $i \in \mathcal{I}$ is in basket $w \in W$, else 0

S_k Capacity (seats) of each airplane in fleet k

d_{wc} Demand for class c in basket w

First-stage input:

x_{nmk} Fleeted schedule provided by the MSPF

Decision variables:

y^r_{ic} Allocation of capacity (seats) to class c on itinerary i

CAM:

$$f_{rev}(x) = \max \sum_i \sum_c v_{ic} y^r_{ic} \tag{6.2a}$$

$$\text{s.t.} \quad \sum_i \sum_c \sigma_{nmi} y^r_{ic} \leq \sum_k S_k x_{nmk} \quad \forall\, (nm) \in \mathcal{A}, i(n) \neq i(m) \tag{6.2b}$$

$$\sum_i y^r_{ic} \eta_{iw} \leq d_{wc} \quad \forall\, w \in W, c \in C \tag{6.2c}$$

$$y^r_{ic} \geq 0 \quad \forall\, i \in \mathcal{I}, c \in C \tag{6.2d}$$

The objective function aims to maximize revenue according to an estimation of ticket sales. Constraints (6.2b) prevent the number of seats on any leg from exceeding the available capacity, according to the size of the assigned fleet type. Equations (6.2c) restrict the capacity allocation to the demand.

6.2.1 Stochastic model

While the average fare paid per class may not vary much for a given time period, the ticket sales mix may differ considerably from day to day. For example, more passengers travel on Sundays and Fridays than on Tuesdays and Wednesdays, as evidenced by major airlines' published flight schedules and fares. A quick check of flight and fare availability in the "Shortcut to Low Fares" section of the Southwest

Airlines website shows that discount fares between two randomly selected cities (Tucson, AZ and Baltimore, MD) are generally more readily available on Monday and Tuesday flights than on Fridays and Sundays. Since travel volumes and the fare class mix can vary from day to day for the same itinerary, we will now model the demand as random data, denoted by $\tilde{d}_{wc}$.

We can employ the L-shaped method of Van Slyke and Wets (1969) to solve the model that includes discretely distributed random demand data. The sample space of this distribution includes S possible outcomes of ticket sales by class. Each scenario $s \in S$ represents a specific demand outcome in each fare class for each basket w. The demand outcome for each basket and class in scenario s is d^s_{wc}. The probability of observing outcome s is p_s. The model for scenario s is

$$h_s(y^{rs}) = \max \sum_i \sum_c v_{ic} y^{rs}_{ic} \tag{6.3a}$$

$$\text{s.t.} \quad \sum_i \sum_c \sigma_{nmi} y^{rs}_{ic} \leq \sum_k S_k x_{nmk} \quad \forall\, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{6.3b}$$

$$\sum_i y^{rs}_{ic} \eta_{iw} \leq d^s_{wc} \quad \forall\, w \in W, c \in C \tag{6.3c}$$

$$y^{rs}_{ic} \geq 0 \quad \forall\, i, c \tag{6.3d}$$

The revenue contribution to MSPF objective function (6.1a) now becomes $\max \sum_s p_s h_s(y^{rs})$, which maximizes the expected value of CAM given numerous possible demand scenarios s. By accounting for variability in demand, we may develop a schedule that might have a lower expected revenue than one based on fixed demand data, but which is more robust to fluctuations in ticket sales.

6.3 Implementation of the revenue approximation models

To demonstrate the utility of MSPF and CAM in finding schedules that maximize revenue, we began by developing additional data to supplement the five-airport, three-day test networks used in previous chapters, and then tested the models with larger networks. We then observed the tradeoffs between fuel costs and revenue in the search for the most profitable schedule in each case. We omitted crew costs from this experiment for simplicity, though an estimate of crew costs per flight could be added to the fuel costs to increase the accuracy of the total costs.

6.3.1 Data generation

The MSPF and CAM required some different data from the other subproblems. For example, previously-generated travel times between airports were multiplied by hourly fuel burn rates for each fleet type to estimate fuel requirements for each flight and fleet. To obtain fuel costs, these fuel requirements were multiplied by a fuel cost per gallon. Note that the fuel cost per gallon could be treated as a random variable, and should be considered random in real-world applications. However, we chose a fixed cost per gallon to simplify the problem for the purposes of this demonstration. Also for simplicity, we chose to use two fleets, with capacities of 130 and 160 seats each. The number of seats per plane in each of the two fare classes was not specified, reflecting a single cabin in each plane in which passengers from both fare classes travel. Demand for each class and basket were generated randomly according to a process described in detail in Appendix C. This process created a matrix of discrete demand levels and probabilities which were referenced for sampling in the model implementation.

6.3.2 Experiment

After generating the required additional data described above, we chose five small test networks from those developed for the experiments for other subproblems. First, we solved each network without randomness to establish baseline objective values and schedules. We then applied the L-shaped method to solve MSPF with the CAM subproblem and random demand data for each network, and recorded the final objective values for the master and subproblems, as well as the schedule solution from the master problem. Note that the initial solution of the master problem is zero in each case, as the objective value for minimizing fuel costs without revenue considerations is zero, and the resulting schedule is also null.

For each test network, we solved the model with 100 scenarios, and refined the objective value using the confidence interval described in Section 3.4. We then performed the same confidence interval procedure with the schedules based on the expected demand, represented as "MV" for mean value in Table 6.1. We also adjusted the tolerances for integrality and optimality to 2% and 1%, respectively, trading some solution precision for considerably shorter computational times. The 1% optimality gap shortened run times from days to less than an hour, in most cases.

We compare the objective values as well as the resulting schedules, using the flight counts and tallies defined in section 3.4. The objective value results are shown in Table 6.1, with values rounded to the nearest thousand. Abbreviation "RD" denotes random demand considered, and LCL and UCL are the lower and upper confidence level of the interval, respectively. A sample flight tally for test case 1's morning flights with fleet 1 appears in Table 6.2, and the flight count comparison tables appear in Appendix D.

		Objective value			
Net	**Cities**	**RD:LCL**	**RD:UCL**	**MV:LCL**	**MV:UCL**
1	5	22,677	22,712	22,519	23,042
2	5	22,957	22,980	22,605	23,307
3	5	20,508	20,809	19,920	20,492
4	5	19,254	19,404	19,289	19,583
5	5	24,540	24,823	24,653	25,152

Table 6.1: Revenue approximation objective values in thousands of units.

In this experiment, the objective values of the deterministic (expected demand) and stochastic (random demand) problem of each test case were quite close. The confidence intervals for the expected demand objective value were either lower than

FLEET 1	**Mean value**					**Randomness**					**Difference**				
City	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1		1	6	0	1		0	4	0	0		1	2	0	1
2	0		0	0	1	0		0	0	3	0		0	0	-2
3	3	0		1	0	1	1		0	2	2	-1		1	-2
4	1	0	0		0	1	0	0		0	0	0	0		0
5	1	0	2	0		2	0	3	0		-1	0	-1	0	

Table 6.2: Sample flight schedule comparison tally: Test case 1, morning flights, fleet 1.

or not statistically significantly different from those that included the consideration of random demand. Therefore, we cannot conclude that the expected revenue increases or decreases when random demand is considered.

Comparisons of the flight schedules with and without randomness seen in Appendix D are somewhat more interesting. The percentage of flights that remained exactly as scheduled is relatively low when examined by fleet, ranging from 7.5% to 30.5%. However, if we include flights that were shifted one or two periods later, the ranges increase to 25% to 70%. If we disregard fleet assignment, the schedules are much more similar, with 75% to 85% of flights not changing or shifted slightly. Therefore, we observe that the introduction of randomness into the problem triggered a considerable change in the fleet assignments. However, the total proportion of flights assigned to each fleet changed little. In test problem 5, 7 flights were shifted from fleet 1 to fleet 2 out of 108 total. In test problem 2, the total fleet assignment was changed by only one flight. The other cases ranged between these two.

Another way to examine the differences in the flight schedules is by the change in service offered in each basket. In most cases, the level of nonstop service offered between any two cities in a basket remained the same or changed by only one or two flights. However, a few cases of larger shifts arose as well. The difference between the number of flights offered from one city to another in the mean value problem and the same sum in the problem with random demand ranged from 11% to 31%

of the total number of scheduled flights. We can infer that the shifts in service occurred in response to the change in demand when randomness was introduced into the problem, so that the schedules produced by the problem with randomness better allow for fluctuations in demand from a revenue-generation standpoint.

The extended ground arcs (EGAs) changed inconsistently with the introduction of random demand. In some cases, the schedule with random demand considered had more consecutive ground arcs, and in other cases the number was reduced. The EGAs sometimes appeared to be shifted from one fleet to another. Since EGAs were not explicitly rewarded in the model, the inconsistency is not surprising, as the model attempts to maximize revenue without aiming to increase robustness to schedule disruption.

6.4 Limitations and extensions

The representation of revenue generation described above aims to estimate the revenue from a given schedule for planning purposes. CAM uses estimated demand data that should reflect the predicted final ticket sales for each flight, but does not maximize revenue in the manner of a bid-price model. Other models exist specifically for revenue maximization, accounting for the dynamic ticket sales process. Such models change the fares and seat allocations by class at different times before departure, and are too complex to represent in a surrogate model.

The models presented are intended to develop an estimation of the revenue that may be generated by a schedule, assuming that some decisions may be performed during the ticket sales process. In other words, we assume that the estimated demands are the result of adjustments made periodically during the ticket sales process, and that these values are more accurate than demand estimated only once when tickets first go on sale. This assumption accounts for the reoptimization performed by most airlines as ticket sales progress. By constraining seat allocations to estimated ticket sales, the model prevents an overoptimistic calculation in which all seats are sold to the most expensive fare class. While the model will allocate seats to the most expensive class first, it may only assign d_{wc} seats to that class, and then must assign seats to a lower fare class. The resulting estimate is passed back to the master of MSP and used to shape the schedule to become more profitable; therefore, consideration of bid-pricing is less important than developing a reasonable estimate of actual revenue. When the MSP must attempt to maximize revenue and minimize fuel and crew costs, schedule tradeoffs will be required to arrive at the most profitable schedule, as we will see in the next chapter.

CHAPTER 7

Integrated Airline Scheduling

In the previous chapters, we have examined several subproblems of the airline planning problem, and developed approximations that allow us to rapidly evaluate a schedule based on contributions from the subproblems. We will now synthesize the integrated problem and explore its behavior. By simultaneously considering the surrogate subproblems in the MSP, we can find a schedule closer to the overall optimal than one found by solving the subproblems sequentially.

7.1 Model structure

The structure of the integrated problem requires a few assumptions. First, we will assume that maintenance is not fleet-specific. In reality, maintenance crews and facilities may work on a few different airplanes within a class (for example, Boeing jets), but airlines require separate training and parts for maintaining different fleet types. By assuming non-fleet-specific maintenance, we may treat maintenance as a single subproblem that includes all fleets. In a real-world application of these techniques, an actual airline may benefit from solving the maintenance subproblem separately for each fleet, to account for fleet-specific maintenance locations and crews.

As discussed in the previous chapter, revenue estimation should be handled with a single subproblem, allowing different fleets to serve passengers wishing to travel in a given basket. However, crew costs are best analyzed with a separate subproblem for each fleet. Each fleet has its own crews and crew pay scale, and each fleet's network requires crew flow conservation and FAA rule compliance. If two similar fleets share crews (such as 757 and 767), we can combine those fleets in a single subproblem. By considering the SCSP separately for each fleet, we may solve a set of smaller crew scheduling problems instead of one large one, and can thus avoid issues that arise with large, sparse matrices.

To implement the integrated problem, we must revisit the MSPF and adjust it to account for all of the subproblems. We can then apply techniques that allow us to solve the problem.

7.2 The suite of models

We must first update MSPF to reflect the cost approximations presented for fleeting, revenue, and crew scheduling, as well as maintenance feasibility, formerly denoted as $\sum_b f_b(x)$ in equation (6.1a). The objective function is now as seen in (7.1a), and constraint (7.1f) is added to require maintenance feasibility. The subproblem models require little modification for incorporation into the integrated MSPF, though all variables x_{nm} will be indexed by fleet as well as origin and departure node. The variables y_{nm} in SCSP must also be indexed as shown in (7.1a) to preserve crew flow by fleet. Fleet-specific pay scales may also be considered as shown. Overnight costs are the same for all fleets, so costs c^o_{nm} are not indexed by fleet.

The complete set of models for integrated MSPF appears as follows. The models use the set and variable definitions presented in previous chapters, except where noted.

Master Schedule Problem with Fleeting (MSPF):

$$\max f_{rev}(x) - \sum_{(nm)\in\mathcal{A}} \sum_{k\in K} c^{fuel}_{i(n)i(m)} x_{nmk} - f_{crew}(x) \tag{7.1a}$$

$$\text{s.t.} \quad \sum_{m} \sum_{k\in K} x_{nmk} \le r^-_{i(n)} \qquad \forall\, n \in \mathcal{N}, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{7.1b}$$

$$\sum_{n} \sum_{k\in K} x_{nmk} \le r^+_{i(m)} \qquad \forall\, m \in \mathcal{N}, (nm) \in \mathcal{A}^{\mathcal{F}} \tag{7.1c}$$

$$\sum_{n} \sum_{k\in K} x_{nmk} \le g_{i(m)} \qquad \forall\, t, m \in \mathcal{N}(t), (nm) \in \mathcal{A}^{\mathcal{G}} \tag{7.1d}$$

$$\sum_{(nm)\in\mathcal{A}(0)} x_{nmk} \le V_k \qquad \forall\, k \in K \tag{7.1e}$$

$$\sum_{(nm)\in\mathcal{A}(0)} \sum_{k\in K} x_{nmk} \le f_{mx}(x) \tag{7.1f}$$

$$\sum_{(mn)\in\mathcal{A}^{\mathcal{F}}, \tau(p)-\gamma(i(n))<\tau(n)\le\tau(p)} x_{mnk} \le x_{pqk} \qquad \forall\, (pq) \in \mathcal{A}^{\mathcal{G}}, k \in K \tag{7.1g}$$

$$\sum_{m} x_{nmk} - \sum_{m} x_{mnk} = 0 \qquad \forall\, n \in \mathcal{N}, k \in K \tag{7.1h}$$

$$x_{nmk} - x_{n'm'k} = 0 \qquad \forall\, k \in K, (nm) \in \mathcal{A}, \tau(n') = \tau(n) + c, \; i(n) = i(n'), i(m) = i(m') \tag{7.1i}$$

$$x_{nmk} \in Z^+ \qquad \forall\, (nm) \in \mathcal{A}, k \in K \tag{7.1j}$$

Surrogate Maintenance Feasibility Problem (MX):

Decision variables:

y^m_{nm} Flow on maintenance arc $(nm) \in \mathcal{A}^{mx}$

$$f_{mx}(x) = \max \sum_{(nm)\in\mathcal{A}^{mx}} y^m_{nm} \tag{7.2a}$$

$$\text{s.t.} \sum_{(lm)\in\mathcal{A}^{mx}(\tau(n))} y^m_{lm} \leq h_{i(n)} \quad \forall\, n \in \mathcal{N}^{mx} \tag{7.2b}$$

$$\sum_{(lm)\in\mathcal{A}^{mx}(\tau(n))} y^m_{lm} + z_{np} = x_{np} \quad \forall\, n, (np) \in \mathcal{A}^{\mathcal{G}}, i(n) \in \mathcal{I}^{mx} \tag{7.2c}$$

$$y^m_{nm} \geq 0 \quad \forall\, (nm) \in \mathcal{A}^{mx} \tag{7.2d}$$

$$z_{nm} \geq 0 \quad \forall\, (nm) \in \mathcal{A}^{\mathcal{G}} \tag{7.2e}$$

Surrogate Crew Scheduling Problem (SCSP):

Data:

C_k Number of crews available for fleet $k \in K$

Decision variables:

y^b_{nmk} block time flow on leg $(nm) \in \mathcal{A}^{\mathcal{F}}$ for fleet $k \in K$

y^s_{nmk} Sit time flow on leg $(nm) \in \mathcal{A}^{\mathcal{G}}$ for fleet $k \in K$

y^d_{nmk} Deadhead flow on $(nm) \in \mathcal{A}^{\mathcal{F}}$ for fleet $k \in K$

y^o_{nmk} Overnight flow on $(nm) \in \mathcal{A}^{\mathcal{O}}$ for fleet $k \in K$

$$f_{crew}(x) = \min \sum_{k \in K} \sum_{(nm) \in \mathcal{A}^{\mathcal{F}}} c^f_{nmk}(y^b_{nmk} + y^d_{nmk}) \quad + \sum_{k \in K} \sum_{(nm) \in \mathcal{A}^{\mathcal{G}}} c^s_{nmk} y^s_{nmk} + \sum_{k \in K} \sum_{(nm) \in \mathcal{A}^{\mathcal{O}} : i(n) \notin I^H} c^o_{nm} y^o_{nmk} \tag{7.3a}$$

$$\text{s.t.} \quad \sum_{l: lm \in \mathcal{A}^{\mathcal{O}}} y^o_{lmk} + \sum_{l: lm \in \mathcal{A}^{\mathcal{G}}} y^s_{lmk} = \sum_{n: mn \in \mathcal{A}^{\mathcal{F}}} (y^b_{mnk} + y^d_{mnk}) + \sum_{n: mn \in \mathcal{A}^{\mathcal{G}}} y^s_{mnk} \quad \forall m \in \mathcal{N}, \forall k \in K \tag{7.3b}$$

$$\sum_{l: lm \in \mathcal{A}^{\mathcal{F}}} (y^b_{lmk} + y^d_{lmk}) + \sum_{l: lm \in \mathcal{A}^{\mathcal{G}}} y^s_{lmk} = \sum_{n: mn \in \mathcal{A}^{\mathcal{G}}} y^s_{mnk} + \sum_{n: mn \in \mathcal{A}^{\mathcal{O}}} y^o_{mnk} \quad \forall m \in \mathcal{N}, \forall k \in K \tag{7.3c}$$

$$\sum_{(lm): \tau(l) \le t, \tau(m) > t} (y^b_{lmk} + y^d_{lmk} + y^s_{lmk}) \le \sum_{(mn): \tau(m) - t^{dp} \le t < \tau(m)} y^o_{mnk} \quad \forall t, \forall k \in K \tag{7.3d}$$

$$\sum_{l: (lm) \in \mathcal{A}^{\mathcal{O}}} y^o_{lmk} = \sum_{n: (mn) \in \mathcal{A}^{\mathcal{D}}} \delta_{mnk} \quad \forall m \in \mathcal{N}, \forall k \in K \tag{7.3e}$$

$$\sum_{l: (lm) \in \mathcal{A}^{\mathcal{D}}} \delta_{lmk} = \sum_{n: (mn) \in \mathcal{A}^{\mathcal{O}}} y^o_{mnk} \quad \forall m \in \mathcal{N}, \forall k \in K \tag{7.3f}$$

$$y^b_{nmk} = x_{nmk} \quad \forall (nm) \in \mathcal{A}^{\mathcal{F}}, \forall k \in K \tag{7.3g}$$

$$y^d_{nmk} \le H x_{nmk} \quad \forall (nm) \in \mathcal{A}^{\mathcal{F}}, \forall k \in K \tag{7.3h}$$

$$\sum_{(mn) \in \mathcal{A}^{\mathcal{D}}(0)} \delta_{mnk} + \sum_{(mn) \in \mathcal{A}^{\mathcal{O}}(0)} y^o_{mnk} \le C_k \quad \forall k \in K \tag{7.3i}$$

$$\delta_{nmk} \ge 0 \quad \forall (nm) \in \mathcal{A}^{\mathcal{D}}, \forall k \in K \qquad y^b_{nmk}, y^d_{nmk} \ge 0 \quad \forall (nm) \in \mathcal{A}^{\mathcal{F}}, \forall k \in K$$
$$y^s_{nmk} \ge 0 \quad \forall (nm) \in \mathcal{A}^{\mathcal{G}}, \forall k \in K \qquad y^o_{nmk} \ge 0 \quad \forall (nm) \in \mathcal{A}^{\mathcal{O}}, \forall k \in K \tag{7.3j}$$

Capacity allocation model (CAM):

Decision variables:

y^r_{ic} Allocation of capacity (seats) to class c on itinerary i

$$f_{rev}(x) = \max \sum_i \sum_c v_{ic} y^r_{ic} \tag{7.4a}$$

$$\text{s.t.} \quad \sum_i \sum_c \sigma_{nmi} y^r_{ic} \leq \sum_{k \in K} S_k x_{nmk} \quad \forall\, (nm) \in \mathcal{A}, i(n) \neq i(m) \tag{7.4b}$$

$$\sum_i y^r_{ic} \eta_{iw} \leq d_{wc} \quad \forall\, w \in W, c \in C \tag{7.4c}$$

$$y^r_{ic} \geq 0 \quad \forall\, i \in \mathcal{I}, c \in C \tag{7.4d}$$

As noted previously, most of the subproblem models are the same as they were when considered individually. We may want to include some slight modifications, such as a "safety factor" in the maintenance surrogate model. We may account for this optional factor by adding a constant required number of surplus maintenance arcs θ in (7.1f):

$$f_{mx}(x) \geq \sum_{(nm) \in \mathcal{A}(0)} \sum_{k \in K} x_{nmk} + \theta \tag{7.5}$$

Alternatively, we can use multiplier $\alpha > 1$ on the number of planes in the schedule:

$$f_{mx}(x) \geq \alpha \sum_{(nm) \in \mathcal{A}(0)} \sum_{k \in K} x_{nmk} \tag{7.6}$$

Given the suite of models, we will consider solution techniques that allow us to find an optimal master schedule.

7.3 Solution techniques

As in previous sections, we can apply Benders' decomposition to solve MSPF, but with with a slight modification. Instead of adding a single cut per iteration, we will add a cut from each subproblem at each iteration, which requires three free variables. These free variables, η_{mx}, η_{crew}, and η_{rev}, represent the contributions of $f_{mx}(x)$,$f_{crew}(x)$, and $f_{rev}(x)$, respectively. The system of equations (A.3a)-(A.3f) are adjusted to include the three subproblems $b \in B$ as follows:

$$\min\ cx + f_{crew}(x) + f_{rev}(x) \tag{7.7a}$$

$$\text{s.t. } Ax = b \tag{7.7b}$$

$$f_{mx}(x) \geq V \tag{7.7c}$$

$$x \geq 0 \tag{7.7d}$$

$$f_j(x) = \min\ q_j y_j \tag{7.7e}$$

$$\text{s.t. } W_j y_j = r_j - T_j x \tag{7.7f}$$

$$y_j \geq 0 \tag{7.7g}$$

$$j \in mx, rev, crew \tag{7.7h}$$

In (7.7a), we substitute $\sum_{b \in B} \eta_b$ for $\sum_{b \in B} f_b(x)$ upon the creation of cuts. Since all of our objective value components are in terms of dollars, we will weight the η_b values equally. In other enterprise models, weighting coefficients may be required to balance the contributions of the subproblems.

The algorithm for solving the integrated model is as follows.

Step 0: Initialize all variables, including iteration counter $k = 0$; partition the problem into master and subproblems

Step 1: Solve MSPF and obtain a schedule x^k

Step 2: Solve MX, SCSP, and CAM with x^k; create cuts per the Benders' algorithm

Step 3: Add cuts to MSPF

Step 4: Solve MSPF to obtain a new x^k

Step 5: Check termination conditions; if the conditions are met, terminate with optimal solution, else increment k and return to Step 1

As in most iterative solution techniques, the termination conditions allow the algorithm to terminate if the difference between the lower and upper bound on the objective value is within a specified tolerance, which may be zero.

The resulting schedule is optimal relative to the surrogate subproblems, and representations of the subproblems can be used to plan the actual maintenance schedule, crew schedule, and capacity allocation. Models that focus specifically on maintenance, crew scheduling, and capacity allocation should be solved with the schedule from MSPF to devise operational plans.

7.4 Implementation of the integrated models

To demonstrate the integrated MSPF, we combined the master problem and the three subproblems for the same collection of 5-airport test networks used in previous chapters, with two fleets per problem. Using specialized code for Benders' decomposition with three subproblems, we solved each test case as described above. Results from these test cases appear in Table 7.1.

			Obj values - integrated model			**Fuel**
Net	**Cities**	**Mx Opps**	**Crew**	**Revenue**	**Total**	**costs**
1	5	21	-615,071	27,725,829	20,644,195	6,466,563
2	5	21	-867,430	28,214,921	21,080,532	6,266,959
3	5	11	-578,674	25,751,074	18,464,252	6,708,148
4	5	23	-575,563	23,574,743	17,906,372	5,092,808
5	5	28	-658,437	29,999,111	21,537,066	7,803,608

Table 7.1: Integrated MSPF experiment results.

The schedule that resulted from each problem accounts for all three subproblems, as well as fuel costs. We compared each of these schedules to the corresponding schedules from the three subproblems using the measures outlines in Section 3.4. The flight count comparison tables appear in Appendix E.

The flight service changes demonstrate that the integrated schedule is aligned much more closely with the optimal schedule for revenue than with the other subproblems. The flight counts and tally tables show that the integrated schedule is very close to the revenue schedule, and quite different from the other schedules. The objective values explain why this effect appears. The revenue objective value comprises a much larger part of the total objective value than the estimated crew costs, and its greater weight clearly overrides the other subproblems in the overall outcome. The maintenance subproblem has very little influence, as it only requires feasibility.

The crew costs are considerably less than the revenue, and it also has much less influence over the integrated schedule. Further, the integrated schedule includes many more flights than the crew-only schedule. Since the crew model involves minimizing crew costs, it tends to reduce the number of flights when the full revenue value of the flights is not considered. The added incentive of revenue drives the integrated model to add many more flights to the schedule. The maintenance model also has little incentive to increase the number of flights, and therefore the maintenance schedules are more limited than the integrated ones.

These observations highlight the need to verify the accuracy of ratios among the subproblem cost coefficients. The test cases used in our experiment included artificial data, and the influence of the relative weight of cost coefficients between the revenue and crew subproblems is readily apparent. When working with real problems, care must be exercised to ensure that the influence exerted by each subproblem on the schedule reflects the true impact of the subproblem on the overall objective value. Though cost coefficients need not be exact, the relationship between them should reflect reality as accurately as possible.

In test case 3, the solution found at termination used fewer than the 15 available aircraft, so that the schedule with only 11 maintenance arcs was still maintenance feasible. This difference seems to result from the problem's network data. Though we would not expect to see unused aircraft in larger, more realistic flight networks, the tradeoffs between the subproblems in this smaller network led to using fewer than the available aircraft used to achieve the best schedule.

7.5 Extensions

Integrated airline scheduling with uncertainty. As seen in investigations of the individual subproblems, consideration of uncertainty builds some flexibility into the airline schedule. To consider randomness in the integrated problem, we may simply include it with the same approaches used in the subproblem discussions, using the L-shaped method instead of Benders' decomposition.

Other costs and subproblems. Our models consider fuel costs, maintenance feasibility, crew costs, and revenue. Many other costs exist in airline operations, such as ground facility operations costs, maintenance costs, and information systems costs for selling tickets. These items can be incorporated in the model set according to an airline's analytical requirements. Perhaps the most interesting additional subproblem would be equipment costs, since lease, purchase, and life-cycle costs (such as maintenance) are major components of an airline's bottom line.

Further applications. In this chapter, we illustrate the use of multiple surrogate subproblems in optimizing a large, complex system. This discussion serves as an initial exposition of the models and some basic solution techniques, and is intended to open further exploration into the concept. While the airline planning problem provides a useful example for illustrating enterprise modelling with surrogates, the paradigm may be readily applied to other large enterprises. Descriptions of some generalized extensions follow in the next chapter.

CHAPTER 8

Summary

In the preceding discussions, we developed and tested surrogate models for three subproblems of the airline planning problem. We demonstrated surrogate model validation with the maintenance feasibility subproblem. In the crew subproblem, we presented models that consider random schedule disruptions. Finally, the revenue subproblem models allowed for randomness in the right-hand side demand data. We then demonstrated an integrated modelling paradigm which employs these surrogates in concert to produce an airline schedule that considers the impact of all of the subproblems. This paradigm can be readily applied to other large, multifaceted enterprises. Such an application requires the development and validation of surrogate representations of the enterprise's subsystems, as well as development of the master problem model to include the contributions of the subproblem contributions to the objective.

8.1 Opportunities for further research

The preceding chapters have concluded with recommendations for extended research. We include a few more here in the context of general enterprise system modelling, and development of novel optimization techniques.

Well-known methods were employed throughout this dissertation to solve the models presented. We applied Benders' decomposition and the L-shaped method, and solved the master and subproblems using CPLEX 8.1. Considerable research has been performed to improve upon the efficiency of CPLEX for specific classes of problems, some of which was discussed in Chapter 2. Some of these methods, particularly those designed for solving large, sparse integer programs, may considerably accelerate the computations required to solve MSP compared to the solution times required by CPLEX. Specifically, column generation may accelerate solution of the MSP master problem, and regularized Benders' decomposition could reduce the time required to solve the entire MSP. Heuristic algorithms that provide a good, though not necessarily optimal, solution may also offer promise. An investigation into the application of advanced solution techniques to the models presented here may enhance the utility of the models, particularly when working with very large systems, including major airlines. Such an investigation should include an implementation of different techniques, and comparison of the performance of these techniques on a collection of both test problems and real problems of various sizes.

The number of subproblems is entirely dependent on the enterprise under consideration. In some systems, these subproblems operate synergistically in ways that do not allow them to be considered separately. Appropriate partitioning of different enterprises into their subproblems, and comparison of results and solution times for alternate subproblem designs, may prove to be an interesting topic of study.

Altering the grouping and number of subproblems changes the number of Benders' cuts per iteration. In Chapter 7, each problem contributed a cut on each iteration. Intuitively, it seems reasonable that allowing each subproblem to contribute a cut on each iteration would trim the feasible region of the master problem less coarsely than a single cut constructed from the results of all of the subproblems. If this concept holds true, then grouping subproblems may reduce the quality of the overall solution. Further, creating a single cut for all subproblems would require a weighting scheme to ensure that the subproblem contributions are prioritized appropriately, if not equally. Research into the effect of multiple cuts from multiple subproblems, versus a single cut per iteration for all subproblems, could advance the state of the art of optimization methods for solving decomposable problems in cases in which subproblems are numerous enough to challenge memory limits. The construction of a single cut per subproblem is preferred when possible for the reasons explained above.

8.2 Conclusion

We have presented a novel modelling paradigm that expands the body of knowledge of techniques for approaching large, complex optimization systems. While considerable opportunity exists to explore the paradigm and its implementation and application, this dissertation has described the concept of enterprise modelling with surrogate subproblems, and has demonstrated the implementation of the scheme in the context of airline planning. The ideas presented here should prove useful in modelling previously intractably large systems, so that they can be optimized in an integrated manner.

APPENDIX A

Computational Methods

Many mathematical programming-based network modelling applications involve hundreds or thousands of nodes and arcs, and therefore hundreds of thousands of variables. Network models generally contain *network constraints*, which involve flow conservation, and *side constraints*, which handle other requirements such as capacity restrictions. The constraint matrices for these problems are usually sparse and highly structured, so special computational methods are used to ensure solution quality and reduce computational time. These methods include column generation and decomposition algorithms.

A.1 Column generation and decomposition

Column generation. The column generation technique was first introduced by Ford and Fulkerson (1958) and applied in a classic paper by Gilmore and Gomory (1961) to solve the cutting-stock problem. The cutting-stock problem involves finding the best way to cut material of a set size into smaller pieces while minimizing waste. Since an immense number of possible ways to cut the material to satisfy demand typically exist, efficient computation is critical. Column generation implicitly considers all possible patterns but only includes a subset in computations.

Dantzig-Wolfe Decomposition. Dantzig and Wolfe (1961) introduce a broadly applicable decomposition algorithm for linear programs of the following form.

$$\min \sum_{i=1}^{n} c_i x_i \tag{A.1a}$$

$$\text{s.t.} \sum_{i=1}^{n} A_i x_i = b \tag{A.1b}$$

$$B_i x_i = b_i \quad \forall\, i \tag{A.1c}$$

$$x_i \geq 0 \quad \forall\, i \tag{A.1d}$$

The iterative method partitions the constraints into a set of *side constraints* (A.1b), which may contain any or all of the variables, and *block constraints* (A.1c), each of which contains a nonoverlapping subset of the variables. These subsets of block constraints are the *subproblems*. The algorithm is based on the fact that any point in a bounded linear program's feasible region can be represented as a convex combination of the extreme points and a nonnegative linear combination of the extreme directions. It uses the side constraints within a *restricted master problem* (RMP), in which only a subset of the extreme points associated with the block constraints are represented. The subproblems are iteratively optimized for each extreme point subset until a global optimal is found.

Dantzig-Wolfe decomposition is applicable to a broad range of problems, including those with a single subproblem. The technique can be described in an economic context where numerous profit-generating units of a business share a market or a set of resources, such that the units operate independently but their actions impact the other units. One example of this paradigm is an airline, in which revenue managers set ticket prices and seat capacity levels, while maintenance schedulers plan major maintenance overhauls for aircraft. Although the actions of revenue management and maintenance scheduling occur independently, the airline's overall profit will suffer if too many jets are scheduled for maintenance at once so that the planned seat capacity is not available for sale.

Recent enhancements. More recent research has refined traditional column generation techniques and addressed problems which may arise in some cases, such as instability and degeneracy. Luebbecke and Desrosiers (2002) discuss current methods for treating such issues. For example, in cases in which the reduced master problem (RMP) is degenerate, Luebbecke and Desrosiers recommend using dual simplex instead of primal simplex. They mention several pricing strategies, including partial pricing, which avoid selection of numerous very similar columns (and therefore can avoid poor matrix conditioning). Their survey also includes a discussion of methods used to alleviate convergence problems, such as stabilized methods involving a bound box, or "trust region," around the incumbent solution that can be moved if a solution on one of the boundaries arises.

Other research focuses on pricing schemes. Mamer and McBride (2000) proposed a decomposition-based pricing procedure that includes all of the original rows and a subset of the columns in the master problem. The dual solution of the master problem is used to construct the subproblem, in which the set of "complicating" constraints are relaxed and moved to the objective function such that violating these constraints incurs a penalty. The new columns of the master problem correspond to nonzeros in the subproblem solution, and the procedure repeats until none of the columns found by the subproblem price out favorably in the master problem. The method aims to maintain a basic feasible solution to the master problem at each iteration, while restricting the size of the problem solved with the simplex method. The algorithm relies on the assumption that a subproblem with a very easily solved structure can be identified.

Column generation can be embedded in other algorithms to improve their solutions. Branch-and-price, discussed by Barnhart et al. (1998), was developed to improve upon the LP bounds of the branch-and-bound solution method used by Gilmore and Gomory to solve the knapsack subproblem of the cutting stock problem. In branch-and-price, columns are priced and added and the LP is resolved using the dual simplex method at each node in the branch-and-bound tree. If no columns price out favorably and no integer solution is found, branching occurs. This technique has been applied successfully to numerous large transportation and communications problems, as noted by Barnhart et al. (1998).

Branch-and-cut is a complementary procedure to branch-and-bound that initially considers a subset of the constraints, and attempts to identify and add valid inequalities to the problem to cut off infeasible solutions found when the LP is solved. Branching occurs when no valid inequalities can be found to cut off an infeasible solution. Several recent research efforts have combined branch-and-cut, also known as row generation, with branch-and-price to form branch-and-price-and-cut. Some of these techniques were derived to exploit the special structure or properties of a specific problem, such as the cutting planes derived by Achuthan et al. (2003) for the capacitated vehicle routing problem.

Benders' decomposition. Kelley (1960) proposes a cutting plane algorithm for approximating convex nonlinear functions, in which a sequence of piecewise linear convex approximations are constructed from the function's supporting hyperplanes. At each step, the newly added supporting hyperplane, or "cut," refines the approximation, until the lower and upper bounds found by the algorithm are close enough to satisfy some stopping criterion. The master problem constraints include the set of cuts, and the master problem is used to find the next point from which a cut is derived. The subproblem identifies the coefficients of the next cut at each step. Benders (1962) builds upon the foundation of Kelley's algorithm to develop a method for solving mixed integer-linear programs (MILP's) with the following structure:

$$\min cx + qy \tag{A.2a}$$

$$\text{s.t. } Ax = b \tag{A.2b}$$

$$Tx + Wy = r \tag{A.2c}$$

$$x, y \geq 0 \tag{A.2d}$$

This LP can be restated as follows:

$$\min\ cx + f(x) \tag{A.3a}$$
$$\text{s.t. } Ax = b \tag{A.3b}$$
$$x \geq 0 \tag{A.3c}$$
$$f(x) = \min\ qy \tag{A.3d}$$
$$\text{s.t. } Wy = r - Tx \tag{A.3e}$$
$$y \geq 0 \tag{A.3f}$$

After finding an initial solution x^k (i.e., $k = 0$), Benders' algorithm creates approximations of the objective value function $f(x)$ defined in (A.3d)-(A.3f) based on the dual statement of the subproblem:

$$\max\ \pi(r - Tx^k) \tag{A.4a}$$
$$\text{s.t. } \pi W \leq q \tag{A.4b}$$

If (A.4) is feasible, the dual solution π^k is used to identify a supporting hyperplane of f at x^k. A cutting plane of the form $\eta \geq \pi^k(r - Tx)$ is then added to the master problem, and the variable η replaces the function $f(x)$ in the objective. If the subproblem is infeasible, the algorithm identifies an extreme ray of (A.4b), μ^k, such that $\mu^k W = 0$, and $\mu^k(r - Tx^k) > 0$. The corresponding cut in this case is $\mu^k(r - Tx) \leq 0$. If the subproblem is feasible, the upper bound for the master problem solution is now $\min\{\{\text{current upper bound}, cx^k + qy^k\}$, where x^k is the solution from the master problem and y^k is the optimal solution of the primal subproblem for iteration k.

The master problem is resolved with the new cut to obtain a new lower bound for the solution. If the difference between the lower and upper bounds is smaller than some preselected tolerance labelled ε, the algorithm terminates. Otherwise, the subproblem is resolved with the new solution x^{k+1} and the algorithm continues.

In some cases and especially when x is high-dimensional, the algorithm tends to jump around the feasible region. This can result in slow convergence to an optimal solution. To reduce this tendency in Benders' decomposition, Ruszczýnski (1986) and others propose methods for *regularizing* the algorithm. Regularizing limits the distance between consecutive solutions $||x^{k+1} - x^k||$, so that the solution "bounces" around the feasible region less and may converge to an optimum more quickly.

Lagrangian relaxation. Some problems can be decomposed into a set of "difficult" and "easy" constraints. For example, in network problems, flow conservation constraints are generally simple, but side constraints can be complex. Consider an LP, $\min\{cx : Ax = b, x \in X\}$, in which constraints $Ax = b$ are "difficult," and $x \in X$ are "easy." The difficult constraints are relaxed and moved into the objective function, where a penalty is incurred if they are violated. Lagrangian relaxation can quickly find good (though not necessarily optimal) solutions, and good bounds on the objective function. It also provides bounds on how far any solution is from optimality.

All of these techniques can be applied to network models, and are particularly useful for multicommodity networks, as these are often decomposed by commodity. Indeed, the multicommodity flow problem discussed by Ford and Fulkerson (1958) was the motivation for the development of Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1961).

A.2 Multicommodity networks

While some networks model the flow of one commodity, such as message packets in a telecommunications network, many other networks handle numerous types of resources. One simple example of a multicommodity flow (MCF) network is the postal system, which handles letters, magazines, small parcels, and large packages. The system must be able to transport all types of mail, but may have specific limitations related to each type - for example, a limited number of large packages can be delivered each day. In planning vehicle capacity, the postal service can decompose the capacity problem into constraints specific to each class of mail, as well as overall capacity constraints. Similarly, MCF network models usually consist of network constraints that can be separated by commodity, and side constraints that incorporate all commodities in joint capacity restrictions.

McBride (1998) presents the MCF problem as follows, where x is the vector of commodity flows:

$$\min\ c^T x \tag{A.5a}$$

$$\text{s.t. } Nx = b \tag{A.5b}$$

$$Ax \leq d \tag{A.5c}$$

$$x \geq 0 \tag{A.5d}$$

The objective function minimizes costs, while constraints (A.5b) are the network constraints and (A.5c) are the joint capacitation and side constraints.

McBride tests an approach called *basis partitioning*, in which he partitions and solves the network constraint subproblems by commodity, and then uses these solutions to solve the full problem. By providing an improved starting basis with a working inverse of a dimensionality at least as small as the number of side constraints, this method can drastically reduce the computational time required to solve MCF problems. If the problem includes many side constraints relative to the number of network constraints, the dimensionality improvement is less pronounced and the computational savings are smaller. McBride's survey compares basis partitioning with decomposition and interior point methods, and finds that his algorithm runs faster than the others for a set of well-known test problems.

Ahuja et al. (1993) discuss two other approaches to solving multicommodity flow (MCF) problems. The first, called *price-directive decomposition*, includes Lagrangian relaxation and Dantzig-Wolfe decomposition and seeks prices that yield an optimal MCF solution. The second method is *resource-directive decomposition*, which approaches the MCF as a capacity allocation problem. Capacity is allocated to individual commodities, and the single commodity flow problems are then solved. The model differs from the basis-partitioning model in that it includes resource allocation variables r_{ij}^k, which represent the number of units of capacity on arc ij assigned to commodity k. Variables x^k and x_{ij}^k represent the total flow of commodity k and flow of k on arc ij respectively, and each arc ij in $\mathcal{A}$ has a capacity of u_{ij} units. $\mathcal{N}$ is the node-arc incidence matrix of the network, and vectors b^k represent the supply and demand of commodity type k at the nodes.

$$\min \sum_k c^k x^k \tag{A.6a}$$

$$\text{s.t. } \mathcal{N} x^k = b^k \quad \forall\, k \tag{A.6b}$$

$$\sum_k r_{ij}^k \leq u_{ij} \quad \forall\, k \tag{A.6c}$$

$$0 \leq x_{ij}^k \leq r_{ij}^k \quad \forall\, A, \forall\, k \tag{A.6d}$$

Constraints (A.6c) ensure that the capacity allocated to each commodity on each arc does not exceed the capacity for that arc, and (A.6d) require that each commodity's flow on an arc does not exceed the capacity allocated to that commodity on that arc. Flow balance is maintained by constraints (A.6b).

Let u_{ij}^k represent the maximum capacity of arc ij that may be allocated to commodity k, where $u_{ij}^k \leq u_{ij}$. The resource-directive method can be solved sequentially, by first assigning the arc commodity capacities r_{ij}^k, and then solving the flow problems. The first step is called the resource allocation problem, and involves minimizing a function $z(r) = \sum_k z^k(r^k)$ subject to (A.6c) and $0 \leq r_{ij}^k \leq u_{ij}^k$ for all arcs (i, j) in A and all commodities k. The function values $\{z^k(r^k)\}$ are known only implicitly, but $z^k(r^k)$ can be found via commodity subproblems as follows:

$$z^k(r^k) = \min\ c^k x^k \tag{A.7a}$$

$$\text{s.t. } \mathcal{N} x^k = b^k \quad \forall\, k \tag{A.7b}$$

$$0 \leq x_{ij}^k \leq r_{ij}^k \quad \forall\, A, \forall\, k \tag{A.7c}$$

Note that $z(r)$ is a piecewise linear convex function.

The MCF models discussed so far have focused on static planning problems, in which capacity and/or flows are optimized. However, many network systems are not static in time. Dynamic networks can be used to model such systems.

A.3 Dynamic networks

Dynamic networks describe network processes or systems that change with time. Each node represents both a location, and a time. Arcs symbolize changes in time and/or location. For example, an emergency evacuation plan may model nodes as locations in a building every fifteen seconds, and the occupants of the building must move from their rooms to an exit without exceeding the capacity of any escape route location at any time. A dynamic network model can allow an emergency planner to minimize the total time needed for all occupants to evacuate the building. If a commodity flowing through a network remains at a location for more than one time period, it flows on a "hold-over" arc, which has end nodes at the same location but at times t and $t + 1$, respectively. Ford and Fulkerson (1962) assert that these dynamic network models are no more complex than static models, but have many more variables and are therefore much larger.

Aronson (1989) presents a survey of dynamic network flow problem applications and algorithms. These methods approach a variety of shortest path, maximal flow, and other network problems as multistage, multiperiod, and other time-indexed contexts largely using the decomposition and solution algorithms described above. Some of the research catalogued by Aronson concerns deterministic problems, and other works acknowledge randomness in stages following the initial step of the problem.

A.4 Randomness in networks

Multicommodity flow networks can be extended to model random behavior of arc capacities, arc costs, and flow requirements. Stochastic programming can be applied to networks to optimize planning decisions when uncertainty is present. For example, Glockner and Nemhauser (2000) introduce a formulation for the dynamic network flow problem with randomness. The model resembles the MCF problem formulation, where commodities correspond to scenarios in the stochastic program. The arc capacities are random in this problem, and arcs are grouped into paths from a single source to a single sink. The solution algorithm is a multistage stochastic program that decomposes the problem by arcs.

The stochastic LP (SLP) is an extension of the deterministic LP in which some of the parameters are random variables. The distribution of the random behavior is derived from past data, forecasts, or other statistical studies, and is treated as an input to the SLP. An appropriate objective function must be selected for each particular problem, such as one of the four common approaches described by Sen and Higle (1999): minimizing expected costs, minimizing absolute deviation from goals (in goal programming), vector optimization under uncertainty, and minimization of maximum costs. In a two-stage SLP, variables are classified as *first-stage* if they correspond to decisions that must be implemented before the outcome of the randomness is observed, or *second-stage* if the corresponding decision may be postponed until after the random events occur. *Multi-stage* problems contain a sequence of decisions and random events, where a *stage* is a time step containing a set of decisions and random events. Decisions made in the second stage or later in response to the outcomes of random events are often called *recourse*. If we can violate these later-stage constraints with some penalty, we encounter *simple recourse*. Examples of simple recourse include backorders in production, in which sales exceed inventory, and overbooking airline flights, where ticket sales exceed seat capacity but passengers can be compensated.

Like simple recourse, general recourse involves problems in which the outcome of the first random event can be observed before the second-stage and later decisions are made. Consider the two-stage general recourse model. Let x_1 represent first-stage variables, or decisions which must be made initially, and let x_{2s} be the second-stage responses to scenario s. Let S represent all possible outcomes of the random events that will occur after the first-stage decisions are made, and let p_s represent the probability of realizing outcome s. The two-stage stochastic LP appears as follows.

$$\min\ c_1x_1 + \sum_{s\in S} p_s c_{2s} x_{2s} \tag{A.8a}$$

$$\text{s.t.}\ A_1x_1 = b_1 \tag{A.8b}$$

$$B_sx_1 + A_{2s}x_{2s} = b_{2s} \quad \forall\, s \in S \tag{A.8c}$$

$$L_1 \le x_1 \le U_1,\ L_2 \le x_{2s} \le U_2 \quad \forall\, s \in S \tag{A.8d}$$

If the matrix A_2 does not vary by scenario, the model has *fixed recourse*.

L-shaped method. Van Slyke and Wets (1969) present the algorithm for solving two-stage SLPs as the *L-shaped method*, which is essentially Benders' decomposition adapted to exploit the separability of the subproblem by scenario. The model can be restated as follows.

$$\min\ cx + E[h_s(x)] \tag{A.9a}$$

$$\text{s.t.}\ Ax = b \tag{A.9b}$$

$$x \ge 0 \tag{A.9c}$$

$$h_s(x) = \min\ q_sy_s \tag{A.9d}$$

$$\text{s.t.}\ W_sy_s = r_s - T_sx \tag{A.9e}$$

$$y_s \ge 0 \tag{A.9f}$$

To calculate $E[h_s(x)]$, we simply multiply $h_s(x)$ for each scenario by the probability of obtaining that scenario, p_s, as seen in equation (A.8a):

$$E[h(x)] = \sum_{s \in S} p_s h_s(x) \tag{A.10}$$

Cuts are generated as $\eta \geq \sum_s p_s \pi_s^k (r_s - T_s x)$ if the subproblem is feasible, and $\sum_s p_s \mu_s^k (r_s - T_s x) \leq 0$ otherwise, where k is the iteration index. These cuts average the optimal dual solutions across all scenarios. As in the Benders' algorithm, the objective function becomes $\min cx + \eta$.

Multicut methods. Ruszczýnski (1986) and Birge and Louveaux (1988) propose methods in which a cut is added for each scenario, instead of creating a single cut as described above. The master problem objective function becomes $\min cx + \sum_{s \in S} p_s \eta_s$. While this method may result in better optimal solutions in cases with few scenarios, the master problem quickly becomes large if many scenarios exist. Birge and Louveaux (1988) discuss methods for clustering the scenarios into sets, which reduces the number of cuts per iteration while retaining advantages of the multicut approach.

Proper modelling of randomness can be critical in many planning problems, especially when the variability significantly impacts decisions. Most systems include numerous random aspects. For example, airlines face potentially disruptive weather patterns, equipment failures, sudden crew illnesses or emergencies which lead to no-shows, air traffic congestion near large airports, and ticket demand fluctuation, among other issues. A thorough understanding of the system in concern is paramount to modelling it as effectively as possible and thus making the best planning decisions.

APPENDIX B

Test Problem Sizes

All test problems span three days incremented by half-hour time periods, for a total of 144 time periods. The revenue approximation subproblem included only one day to limit the problem size, since we assume that the capacity allocation for each day is identical. Therefore, the daily revenue remains the same as well, and the resulting subproblem objective value was simply tripled when included in MSP.

Net	Airports	Planes	Fleets	Rows	Columns	Subproblems
1-5	5	15	1	3,600	6,121	Mx, Crew
1-5	5	15	2	7,200	12,242	Revenue, Integrated
6	20	60	1	72,382	57,980	Mx, Crew
7	25	75	1	108,602	90,600	Mx, Crew
8	30	90	1	152,072	130,470	Mx, Crew

Table B.1: MSP master problem sizes.

Network	Airports	Mx stations	Rows	Columns
1,3-5	5	2	576	576
2	5	3	864	864
6	20	3	864	864
7	25	3	864	864
8	30	3	864	864

Table B.2: Maintenance subproblem sizes.

In the maintenance subproblem, note that the size depends on the number of maintenance stations. Therefore, a large network with three maintenance stations may have the same size subproblem as a smaller network with two maintenance stations.

Network	Airports	Rows	Columns
1-5, single fleet	5	8,786	45,360
1-5, two fleets	5	14,689	90,720

Table B.3: Crew subproblem sizes.

Network	Airports	Rows	Columns
1-5, two fleets	5	1,080	8,736

Table B.4: Revenue subproblem sizes.

The revenue subproblem spanned only one day of each schedule in order to limit the problem size. The objective value was multiplied by the number of days in the master schedule. This simplification follows the assumption that both the schedule and the capacity allocation remain static from day to day. In cases where the schedule is allowed to differ on some days, such as weekends, additional columns must be added to the revenue subproblem to find the best capacity allocation for those days, and the factors on the revenue coefficients in the objective function must be adjusted accordingly.

The integrated models include the master problem and each of the subproblems. Therefore, the small network problems have 28,588 rows and 136,032 columns in most cases (some slight variations result from different model parameters).

APPENDIX C

Random Data Generation for Model Implementation

The following discussions detail the processes used to develop distributions and data for the test problems used in the surrogate model implementation experiments. The resulting distributions are coarse representations of the real random processes. More realistic data could be constructed by collecting the relevant information about real flight networks, and performing statistical analyses to estimate the actual distributions. Clearly, the actual distributions and the selection of discretization levels depend on the specific flight network under study.

C.1 Maintenance scheduling: a surrogate representation

To generate problem data, we begin with networks of 5 airports over 144 thirty-minute time periods, or three days, with a maximum of 15 aircraft in each schedule. For each network, we used a pseudo-random number generator to produce grid coordinate pairs for each city. The coordinates were uniformly distributed on the interval [100,3000] in the east-west direction and [100,1500] in the north-south direction. From the coordinates, we calculated Euclidean distances between the cities, and linearly transformed these distances into matrices of travel times. We then randomly generated profit per flight and demand per flight ($\{c_{ij}\}$ and $\{d_{ij}\}$, respectively) using uniform distributions, and selected airport capacities and number of maintenance crews at each airport. Initially, each airport was identified as a maintenance station with probability 0.2. Maintenance stations are then designated sequentially, although this designation is adjusted to ensure that at least one, and no more than two, airports are designated as maintenance stations. Within each maintenance station, there are 2, 3, or 4 maintenance crews with probabilities 0.1, 0.2, and 0.7, respectively.

C.2 Surrogate crew schedule problem (SCSP) with disruption

The data sets for random schedule disruptions include arbitrary mean disruption probabilities based on time of day and location. Some airports were arbitrarily selected to have higher mean probabilities of disruption upon departure and/or arrival, and the mean probability of disruption increased from morning to evening for all locations. We used four time windows: 5am-12pm; 12pm-5pm; 5pm-10pm; and 10pm-5am. These periods represent morning, afternoon, evening, and night. The mean disruption probabilities were then adjusted by a uniform randomly generated value within a percentage increment of the mean value for each location and time window, where the percentage increment for short delays was smaller than for longer delays. Delays of 1, 2, or 3 time increments were included for each location and time window.

C.3 Revenue approximation with random demand

Random demand for CAM was generated using several arbitrary discrete distributions, as described below. These distributions are artificial and designed to resemble reality, but are much more simplistic than distributions that could be constructed using data from a real airline. The data are intended for use only as demonstrative test cases.

To simulate passenger demand for each class and O-D pair in each of three time windows, we generated a uniform random variable between 0 and 1 and compared each outcome to an arbitrary discrete distribution to find a demand difference from the fixed demand value of 1.5 flights. Table C.1 contained this discrete distribution, and was referenced to select the variation from the average demand on each leg. For example, if the outcome of the uniform random variable generation fell between 0.45 and 0.75, a value of 1 flight was added to the fixed demand of 1.5 flights to produce a 2.5-flight demand. This value was then multiplied by a selected number of seats for each class, with less demand for the more expensive classes. The resulting

Outcome	Demand difference
0 - 0.25	0
0.25 - 0.45	1
0.45 - 0.75	2
0.75 - 0.95	3
0.95 - 1.0	4

Table C.1: Sample demand generation discrete distribution

demand for seats per basket and class was then recorded as the fixed demand d_{wc} for the basket and class in question.

To generate three scenarios per wc combination, the d_{wc} values were multiplied by 0.6 and added to a random variable with a uniform distribution ranging between 0% and 80% of the fixed demand value. Therefore, the demand scenarios for each wc combination were often quite different, reflecting the fickleness of the market and variability in demand on different days of the week. The distributions used for generating these scenarios, as well as the number of scenarios, could be validated and fine-tuned to improve accuracy in the model, but such an investigation is beyond the scope of this study.

Itineraries for each basket were created in two steps, where a maximum of one connection was allowed per passenger itinerary. In the first step, we used a uniform random variable and an arbitrary discrete distribution to choose whether zero, one, or two connecting city options would be allowed for each O-D pair. If this outcome assigned no connecting cities, we allowed only direct flights between the O-D pair. An outcome of one or two connecting city options required the use of another uniform random variable to select each connecting city. A direct flight was also a possible choice. If the same connection city was chosen in cases of two allowable connection cities, the outcome was discarded and reselected until a different city or a direct flight was selected. For example, assume that we selected two connecting cities for travel from city B to city D in a 5-city network. Our first outcome permitted connections in city C. If our second outcome also selected city C, we then reselected until we chose A, E, or a direct flight (city 0).

The second step of itinerary generation required the use of minimum and maximum allowable layover lengths. For each O-D pair and connection city, we began by assigning a flight departing from the origin city for the connection city at the earliest hour in the basket w to a new itinerary, and assigned an itinerary index. Unless this itinerary was a direct flight, we then assigned the flight departing from the connecting city after the first flight's arrival time, plus the minimum layover time, to the itinerary. We then repeated this step for each connecting flight departing for the destination at each time increment from the minimum to the maximum layover time, assigning each itinerary its own index. The process was repeated with the origin-connection flight departing at each time increment in the time window, as well as each selected connection city. The entire procedure was then repeated for each basket.

APPENDIX D

Revenue Approximation: Schedule Comparison Results

As described in Section 3.4, the tables below contain the following metrics for comparing two schedules:

- The number of flights that appear in both schedules
- The number of flights that are nearly the same in both schedules, defined as flights that share an O-D pair but fly two or fewer time increments later in the second schedule
- The number of "extended" ground arcs (EGAs), or consecutive ground arcs that exceed the minimum turn time in each schedule, as a measure of schedule robustness
- The total number of flights in each schedule

For each test case, the most profitable schedule for mean value demand (MV) is compared with that for random demand (RD).

	Identical	"Near"	EGAs		Total Flights	
Fleet	flights	flights	MV	RD	MV	RD
1	15	8	123	99	52	53
2	7	14	72	81	49	54
Both	49	36	195	180	101	107

Table D.1: Test case 1, flight count summary.

Fleet	Identical flights	"Near" flights	EGAs		Total Flights	
			MV	RD	MV	RD
1	9	1	99	84	32	33
2	17	5	78	54	33	32
Both	43	8	177	138	65	65

Table D.2: Test case 2, flight count summary.

Fleet	Identical flights	"Near" flights	EGAs		Total Flights	
			MV	RD	MV	RD
1	3	15	90	78	38	40
2	9	2	51	99	39	36
Both	28	33	141	177	77	76

Table D.3: Test case 3, flight count summary.

Fleet	Identical flights	"Near" flights	EGAs		Total Flights	
			MV	RD	MV	RD
1	9	16	87	90	36	40
2	3	7	54	69	40	40
Both	22	40	141	159	76	80

Table D.4: Test case 4, flight count summary.

Fleet	Identical flights	"Near" flights	EGAs		Total Flights	
			MV	RD	MV	RD
1	18	4	105	132	59	52
2	13	9	84	93	49	56
Both	70	12	189	225	108	108

Table D.5: Test case 5, flight count summary.

APPENDIX E

Integrated Model: Schedule Comparison Results

As described in Section 3.4, the tables below contain the following metrics for comparing two schedules:

- The number of flights that appear in both schedules
- The number of flights that are nearly the same in both schedules, defined as flights that share an O-D pair but fly two or fewer time increments later in the second schedule
- The total number of flights in each schedule

For each test case, the most profitable schedule from the integrated model (Int) that considers all three subproblems (Sub) is compared with that for only maintenance (Mx), crew (Crew), or revenue (Rev). For the revenue subproblems, the fleet schedules are compared separately, and then without regard to fleet assignment (denoted "Both.")

Sub/ Fleet	Identical flights	"Near" flights	Total Flights	
			Sub	Int
MX/Both	3	2	40	87
Crew/Both	2	2	18	87
Rev/1	1	5	52	41
Rev/2	1	4	49	46
Rev/Both	16	21	101	87

Table E.1: Test case 1, flight count summary.

Fleet	Identical flights	"Near" flights	Total Flights Sub	Total Flights Int
MX/Both	2	10	44	64
Crew/Both	3	1	20	64
Rev/1	17	3	32	31
Rev/2	23	3	33	33
Rev/Both	41	10	65	64

Table E.2: Test case 2, flight count summary.

Fleet	Identical flights	"Near" flights	Total Flights Sub	Total Flights Int
MX/Both	1	6	30	72
Crew/Both	1	2	16	72
Rev/1	4	7	38	37
Rev/2	6	1	39	35
Rev/Both	21	17	77	72

Table E.3: Test case 3, flight count summary.

Fleet	Identical flights	"Near" flights	Total Flights Sub	Total Flights Int
MX/Both	0	1	56	77
Crew/Both	2	2	16	77
Rev/1	0	0	36	32
Rev/2	0	6	40	45
Rev/Both	0	10	76	77

Table E.4: Test case 4, flight count summary.

Fleet	Identical flights	"Near" flights	Total Flights Sub	Total Flights Int
MX/Both	11	3	74	95
Crew/Both	3	3	16	95
Rev/1	2	5	59	49
Rev/2	0	6	49	46
Rev/Both	11	23	108	95

Table E.5: Test case 5, flight count summary.

REFERENCES

Achuthan, N., L. Cacetta, and S. Hill (2003). An improved branch-and-cut algorithm for the capacitated vehicle routing problem. *Transportation Science* **37**(2).

Ahuja, R. K., T. L. Magnanti, and J. B. Orlin (1993). *Network Flows*. Englewood Cliffs, NJ: Prentice Hall.

Anbil, R., R. Tanga, and E. Johnson (1992). A global optimization approach to crew scheduling. *IBM Systems Journal* **31**, 71–78.

Andersson, E., E. Housos, N. Kohl, and D. Wedelin (1998). Crew pairing optimization. In G. Yu (Ed.), *Operations Research in the Airline Industry*. Kluwer Academic Publishers.

AP (2004). Airlines could save millions. *CBS News.com*.

Arabeyre, J., J. Fearnley, F. Steiger, and W. Teather (1969). The airline crew scheduling problem: A survey. *Transportation Science* **3**.

Aronson, J. E. (1989). A survey of dynamic network flows. *Annals of Operations Research* **20**, 1–66.

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

Barnhart, C., L. Hatay, and E. L. Johnson (1995). Deadhead selection for the long-haul crew pairing problem. *Operations Research* **43**(3).

Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance (1998). Branch-and-price: Column generation for solving huge integer problems. *Operations Research* **46**(3).

Barnhart, C., T. S. Kniker, and M. Lohatepanot (2002). Itinerary-based fleet assignment. *Transportation Science* **36**(2), 199–217.

Barnhart, C., F. Lu, and R. Shenoi (1998). Integrated airline schedule planning. In G. Yu (Ed.), *Operations Research in the Airline Industry*, pp. 384–403. Kluwer Academic Publishers.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* **4**, 238–252.

Birge, J. R. and F. V. Louveaux (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operations Research* **34**(3), 384–392.

Clarke, L. and P. Gong (1995). Capacitated network design with column generation. Technical report, Georgia Institute of Technology.

Cohn, A. M. and 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, and J. Desrosiers (2001). Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science* **35**(4), 375–388.

Dantzig, G. B. and P. Wolfe (1961). The decomposition algorithm for linear programs. *Econometrica* **29**(4), 767–778.

Ford, Jr., L. and D. Fulkerson (1958). A suggested computation for maximal multicommodity network flows. *Management Science* (5), 97–101.

Ford, Jr., L. and D. Fulkerson (1962). *Flows in Networks*. Princeton: Princeton University Press.

Freling, R., D. Huisman, and A. P. M. Wagelmans (2000). Models and algorithms for integration of vehicle and crew scheduling. Technical Report ERS-2000-14-LIS, Erasmus Research Institute of Management.

Gamache, M. and F. Soumis (1998). A method for optimally solving the rostering problem. In G. Yu (Ed.), *Operations Research in the Airline Industry*, pp. 124–157. Kluwer Academic Publishers.

Gamache, M., F. Soumis, G. Marquis, and J. Desrosiers (1999). A column generation approach for large-scale aircrew rostering problems. *Operations Research* **47**(2).

Gilmore, P. and R. Gomory (1961). A linear programming approach to the cutting stock problem. *Operations Research* **9**(6), 849–859.

Glockner, G. D. and G. L. Nemhauser (2000). A dynamic network flow problem with uncertain arc capacities: Formulation and structure. *Operations Research* **48**(2), 233–242.

Gopalan, R. and K. Talluri (1998). The aircraft maintenance routing problem. *Operations Research* **46**, 260–271.

Graves, G. W., R. D. McBride, I. Gershkoff, D. Anderson, and D. Mahidhara (1993). Flight crew scheduling. *Management Science* **39**(6).

Hane, C. A., C. Barnhart, et al. (1995). The fleet assignment problem: solving a large-scale integer program. *Mathematical Programming* **70**, 211–232.

Higle, J. L. and S. Sen (1997). A stochastic programming model for network resource utilization in the presence of multi-class demand uncertainty. Technical report, University of Arizona.

Jarrah, A. I. (2000). An efficient airline re-fleeting model for the incremental modification of planned fleet assignments. *Transportation Science* **34**(4).

Kelley, J. E. (1960). The cutting-plane method for solving convex programs. *Journal of the SIAM* **8**, 703–712.

Klabjan, D., E. Johnson, G. Nemhauser, E. Gelman, , and S. Ramaswamy (2001). Solving large airline crew scheduling problems: Random pairing generation and strong branching. *Computational Optimization and Applications* **20**, 73–91.

Klabjan, D., E. L. Johnson, G. L. Nemhauser, E. Gelman, and S. Ramaswamy (2002). Airline crew scheduling with time windows and plane-count constraints. *Transportation Science* **36**(3), 337–348.

Kohl, N., A. Larsen, J. Larsen, A. Ross, and S. Tiourine (2004). Airline disruption management - perspectives, experiences and outlook. Technical report, Carmen Systems.

Lettovský, L., E. L. Johnson, and G. L. Nemhauser (2000). Airline crew recovery. *Transportation Science* **34**(4).

Luebbecke, M. E. and J. Desrosiers (2002). Selected topics in column generation. Les Cahiers du GERAD G-2002-64, GERAD, Montréal, Canada.

Mak, W., D. Morton, and R. Wood (1999). Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters* **24**, 47–56.

Mamer, J. W. and R. D. McBride (2000). A decomposition-based pricing procedure for large-scale linear programs: an application to the linear multicommodity flow problem. *Management Science* **46**(5), 693–709.

McBride, R. D. (1998). Advances in solving the multicommodity flow problem. *Interfaces* **28**(2), 32–41.

McGill, J. I. and G. J. Van Ryzin (1999). Revenue management: Research overview and prospects. *Transportation Science* **33**(2), 233–256.

Mingozzi, A., M. Boschetti, S. Ricciardelli, and L. Bianco (1999). A set partitioning approach to the crew scheduling problem. *Operations Research* **47**(6), 873–888.

Pak, K., R. Dekker, and G. Kindervater (2003). Airline revenue management with shifting capacity. ERIM Report Series ERS-2003-091-LIS.

Pak, K. and N. Piersma (2002). Airline revenue management. ERIM Report Series ERS-2002-12-LIS.

Reuters (2004). Airlines raising fares to offset fuel costs. *CNN.com*.

Rexing, B., C. Barnhart, T. Kniker, A. Jarrah, and N. Krishnamurthy (2000). Aircraft fleet assignment with time windows. *Transportation Science* **34**(1).

Rubin, J. (1973). A technique for the solution of massive set covering problems, with applications to airline crew scheduling. *Transportation Science* **7**, 34–48.

Ruszczýnski, A. (1986). A regularized decomposition for minimizing a sum of polyhedral functions. *Mathematical Programming* **35**, 309–333.

Ryan, D. (1992). The solution of massive generalized set partitioning problems in aircrew rostering. *Journal of the Operational Research Society* **43**, 459–467.

Schaefer, A. J., E. L. Johnson, A. J. Kleywegt, and G. L. Nemhauser (2003). Airline crew scheduling under uncertainty. University of Pittsburgh.

Sen, S. and J. L. Higle (1999). An introductory tutorial on stochastic linear programming models. *Interfaces* **29**(2), 33–61.

Stojković, M. and F. Soumis (2001). An optimization model for the simultaneous operational flight and pilot scheduling problem. *Management Science* **47**(9), 1290–1305.

Stojković, M., F. Soumis, and J. Derosiers (1998). The operational airline crew scheduling problem. *Transportation Science* **32**(3).

Subramanian, J., S. Stidham, and C. Lautenbacher (1999). Airline yield management problem with overbooking, cancellations, and no shows. *Transportation Science* **33**(2), 147–167.

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

U.S. House of Representatives (2000). *The Subcommittee on Aviation Hearing on the Effect of Fuel Price Increases on Airlines and Passengers*, Number 106-113. U.S. House of Representatives.

Van Slyke, R. and R. J.-B. Wets (1969). L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics* **17**, 638–663.

Vance, P. H., C. Barnhart, E. L. Johnson, and G. L. Nemhauser (1997). Airline crew scheduling: A new formulation and decomposition algorithm. *Operations Research* **45**, 188–200.

Wedelin, D. (1995). An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Annals of Operations Research* **57**, 283–301.

Yen, J. W. and J. R. Birge (2000). A stochastic programming approach to the airline crew scheduling problem. Technical report, Northwestern University.