INTEGRATED FLEET ASSIGNMENT

WITH CARGO ROUTING

LI DONG

(B. Eng., M. Eng. XJTU)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

Acknowledgement

I would like to express my profound gratitude to my two supervisors, Prof Huang Huei Chuen and Prof Chew Ek Peng, for their invaluable guidance and support during my stay at NUS. Without them this thesis can never be finished.

Special thanks to Prof Ellis Johnson from Georgia Institute of Technology, whose advice makes my research works more meaningful. Thank Dr. Alexander David Morton for his continuous help and advice throughout my research. He is also my good friend who shared with me lots of interesting stories in his hometown.

I extend my gratitude to all my friends who made my stay at NUS an experience I will never forget. Thanks Liang Zhe (best friend in Singapore), Lin Wei (former roommate), Bao Jie, Xu zhiyong (badminton partners), Mok Tsuey Wei, Ivy, Yong Yean Yik, Leong ChunHow, and Cheong Wee Tat (Malaysian friends), and all other students in Ergonomics Lab where I spent most of the time in the past two years.

Finally, thanks my family for their support, understanding, and encouragement throughout the course of my study and research.

Table of Contents

Summary

Fleet assignment is the second airline schedule planning step that aims to maximize the profitability by optimally assigning fleet type to the legs. Traditionally this step ignores the cargo flow completely. As the revenue contributed by cargo keeps increasing for the last decade, the cargo flow is very important for combination carriers and should be properly modeled. The route of cargo is determined to a large extent by the cargo capacity of every leg, which depends on the fleet assignment decision. As a result, the fleet assignment has great influence on the cargo revenue. Incorporating the cargo routing into the fleet assignment can help the combination carrier to better balance its resource and the forecasted cargo demand.

This paper proposes an integration of the fleet assignment model and the cargo routing model. To eliminate the complexity brought by the time window and the side constraints, a two phase technique was applied to formulate the cargo routing problem as a path-oriented multicommodity network flow model, in which each column corresponds to a feasible path. To accommodate the uncertainty of the feasible path, the capacity constraints and the variables in the cargo routing model are disaggregated and all potential feasible paths are generated from a non-fleeted schedule to replace feasible paths.

A Benders decomposition based algorithm is proposed to solve the integrated problem.

This algorithm decomposes the integrated formulation into a relaxed master problem of the fleet assignment and a subproblem of the cargo routing. These two problems are solved iteratively until the difference between their solutions is within a designated tolerance. Other than the basic algorithm, three variants, a pareto-optimal cut generation approach, an ε-optimal approach and a hybrid approach are applied to solve the integrated problem. The pareto-optimal cut generation approach selects strong cuts at each Benders iteration; the ε-optimal approach solves the Benders relaxed master problem only to a feasible integer solution rather than the integer optimum; the hybrid approach first employs the ε-optimal approach to find a good feasible solution, then it turns to the basic algorithm for the real optimum.

Computational results show that the basic algorithm outperforms all the others in our problem. It spent only several minutes to obtain the optimal solution for all test instances. The hybrid approach also converges very fast and has the potential to be an efficient method if an appropriate turning point is selected.

List of Figures

List of Tables

1 Introduction

Airline schedule planning consists of a series of planning activities that have to be made so that the schedule is operationally feasible and profitable. Normally these activities are carried out in a sequential process rather than simultaneously, which reduces the complexity of the planning process, but leaves scope for improvement. To generate improved plans these activities could be integrated together or incorporate other problems that have linkages with them.

The recent advance in the computer hardware and the solution algorithm make possible the simultaneous planning. Many integrated approaches that carried out two or more planning activities simultaneously were developed and improved plans over the sequential approach were reported. The main challenge of the integrated planning is to obtain improved plans in reasonable time.

This chapter introduces the background, motivation and main contributions of this research. Section 1.1 introduces the traditional airline schedule planning, followed by Section 1.2 that presents recent development of the integrated planning approach. Next, the problem we studied is described in Section 1.3 and our main contributions are reported in Section 1.4. This chapter ends with the organization of the rest of this thesis in Section 1.5.

1.1 Traditional Airline Schedule Planning

The flight schedule is the primary product of an airline. It defines a list of legs to be flown by the airline. A flight *leg* is a non-stop flight from an origin to a destination with specific departure and arrival times (Gopalan and Talluri, 1998a). The flight schedule has significant influence on customers' choices and thus profit of an airline. The objective of airline schedule planning is then to find a schedule that best balances customers' demand and an airline's resource.

Airline schedule planning includes all the planning activities that have to be carried out for a schedule to be considered operational and profitable. There are many factors to be taken into account before marketing the schedule, such as flight operations, airport facilities, and regulations of Federal Aviation Administration (FAA) on aircraft maintenance and crew rest time. The main decision problems involved in schedule planning are schedule development, fleet assignment, through flight selection, aircraft routing and crew scheduling, along with the sequence in which the decisions are made. Normally these problems are solved in a sequential process rather than simultaneously, even though simultaneous solution approaches are preferred. The primary reason is because the integrated problem lies beyond the capability of the current computer technology and solution algorithm. These decision problems have been well studied and a number of mathematical models have been developed to solve them individually. A thorough description on airline schedule planning can be found in the survey by Gopalan and Talluri (1998a). Here we only introduce these problems briefly.

1.1.1 Schedule Development

Flight schedules are commonly constructed based on the market demand. The demand of a market between two cities over a period of time is estimated by traffic forecasting. Since the demand fluctuates greatly with lots of factors like the season, the economy status, and even the holidays, traffic forecasting has significant influence on the effectiveness and profitability of a schedule. More accurate traffic forecasting may help the airline to better serve markets by judiciously designing the flight network and assigning frequency to the legs. Currently the flight schedule is developed manually for most airlines. After the schedule, or a list of legs, is determined, the next step is to effectively allocate resources, both aircraft and personnel, to fly the schedule, which is accomplished by the four steps described in the following Sections 1.1.2- 1.1.5.

1.1.2 Fleet Assignment

Given a flight schedule about where and when to fly and different aircraft fleets, the fleet assignment is made to determine which fleet to assign to each leg. The fleet assignment problem is normally modeled as a multicommodity flow problem with the objective to minimize the total assignment cost, which consists of the operating cost, carrying related cost, and spill cost. *Spill cost* of a leg is the loss of revenue when the capacity of the aircraft assigned to it is not enough to satisfy all the customers' demands on this leg. The airline either recaptures the excess demands by other legs in its own network, or spills them to another airline. Three sets of constraints must be satisfied by the fleet assignment solution. Firstly, each leg must be covered once and only once by a fleet type. Then, aircraft going into a station at a particular time must leave the same station at some time later. Finally, the number of aircraft in use in a fleet type can not exceed the size of this fleet. The fleet assignment model is defined on a time space dynamic flight network, where a node represents the time and space, and an arc represents the movement between time and/or space. The model is often applied to the daily fleet assignment that follows the same schedule every day in a week. However, it can be easily extended to a weekly fleet assignment model by building the flight network for a week rather than for a single day. Many research works have been done to study the fleet assignment problem. A detailed literature review is provided in the next chapter.

1.1.3 Through Flight Selection

When traveling from one city to another, passengers will firstly choose the non-stop flights. However, an airline can not serve every market with non-stop flights because of resource limitation. In the absence of such flights, the connection service with two or more legs is provided. As a type of connection, a *through flight* uses the same aircraft for all the legs covered. Passengers prefer through flights because they do not need to leave the plane and locate the departure gate for their next flight through a crowded airport to make a tight connection. Through flights thus have an additional marketing advantage over regular connections. As a result, choosing a profitable set of through flights is an important step in the airline schedule planning. To accomplish this, the benefit of each potential through flight has to be evaluated first. The benefit can be measured in terms of the incremental revenue from making a given pair of legs a through flight. Several requirements must be met for a through flight to be valid. Among of them are the minimum transit time and maximum waiting time onboard between two connected legs, non-circuitous connection and some constraints at the market level. Furthermore, the selection of through flights must be made in such a way that the number of aircraft required to fly the schedule is not increased. For a more detailed description, the reader can refer to previous works by Bard and Cunningham (1987) and Jarrah and Strehler (2000).

1.1.4 Aircraft Routing

After the aircraft fleet on a leg is specified by the fleet assignment, the aircraft routing problem is solved to determine which one of the specific aircraft in that fleet, referred by a tail number, actually flies that leg. In other words, aircraft routing assigns a sequence of legs to each individual aircraft so that appropriate aircraft maintenance is ensured. For safety reasons, every aircraft must regularly go through different types of maintenance checks with different scope, frequency, and duration. For instance, FAA requires *A*, *B*, *C* and *D* checks (Gopalan and Talluri, 1998a). Type *A* checks inspect all the major systems and are performed frequently. If an aircraft does not undergo this check within a specified period, it is forbidden to fly. The aircraft routing problem is solved mainly by two classes of approaches. The first one explores the structure of the underlying lines of flying (LOFs) to derive heuristics (Gopalan and Talluri, 1998b). For the daily aircraft routing, LOFs specify the origin at the start of the day and the destination at the end of the day for a specific aircraft. The other class of approaches uses the mathematical programming models. Kabbani and Patty (1992) proposed a general set partitioning formulation for the aircraft routing problem, with the objective to minimize the total routing cost. This approach generates a large number of feasible maintenance routings and selects a subset of them to cover every leg in the schedule. Therefore, arbitrary and complex maintenance rules can be easily incorporated into the routing selection.

1.1.5 Crew Scheduling

After fuel costs, crew costs are the highest direct operating cost of an airline. The crew scheduling problem partitions a given flight schedule into crew rotations or parings and assigns them to flight crews, such that each pairing satisfies a set of work rules and the total assignment cost is minimized. A *crew paring* consists of a sequence of duty and rest periods, where a duty is a set of legs flown by a crew in a single workday. The duration of a duty, called *duty period*, is usually restricted to eight hours of flying and twelve hours of total duty time, including briefing before the first leg of the duty and debriefing after the last leg of the duty. Between duty periods, there must be enough overnight rests or layovers. Every paring begins and ends at the same crew base and must satisfy the many rules governing the legality and penalty of the pairing, such as the flying time, resting time, connection time, etc. Moreover, the crew pairing problem must take into account the number of crews available at different crew bases, the preference to keep crews on the same plane during a duty period, and the preference for short pairing. One of the related works is from Anbil et al. (1992), who presented a global approach to optimize the crew pairing problem.

For some airlines, after parings are constructed, they are finally assigned to every individual crew by solving a *crew bidding problem* or a *crew rostering problem*. A *bidline* or *roster* consists of a sequence of pairings that a crew flies within a month. Like pairings, bidlines and rosters must observe all relevant rules.

1.2 Integrated Airline Schedule Planning

The sequential approach considerably reduces the complexity of the airline schedule planning process. However, it may not yield feasible solutions due to the reduced flexibility by previous decisions, and even if feasible solutions are obtained, these solutions might be far from optimal. For instance, the solution of the fleet assignment may lead to the resulting aircraft routing problem infeasible, as certain maintenance constraints may not be satisfied. Solving the fleet assignment problem and the crew scheduling problem sequentially may create a situation where the saving in fleet assignment can never compensate the excessive crew costs. As a result, the simultaneous decision making is preferred because it could produce more economical plans and eliminate incompatibilities between decisions. By synchronizing some or all of these decisions, the integrated airline schedule planning reduces the overall schedule generation time and improves the productivity of the schedule developers. However, the development of integrated planning has been limited by the computer hardware and the solution algorithm for many years. Until recently, the integrated planning becomes possible with the great development in both of the two fields. Many integrated approaches that simultaneously solve two or more problems were developed and improved solutions over the sequential approach were reported in the past five years. Some of them simultaneously solve fleet assignment and aircraft routing, fleet assignment and crew scheduling, or crew scheduling and aircraft routing.

Besides integrating with each other, some decision making processes also incorporate other problems that have linkages with them. These problems usually have great influence on the profitability of an airline. Taking the fleet assignment as an example, its solutions determine the seat number of every leg and thus have significant influence on the passenger flow, which is the main source of an airline's revenue. Therefore, a better fleet assignment decision can be expected if these two problems are simultaneously solved.

1.3 Problem Description

The second airline schedule planning step, fleet assignment, aims to maximize profitability by optimally allocating fleet types to legs. Traditionally this step considers only the leg based passenger flow and ignores the cargo flow completely. As all legs in a network are interdependent, failing to capture the network effect may cause the solution to be suboptimal. The incorporation of itinerary based passenger flow into the fleet assignment had been studied by Barnhart et al. (2002). As the revenue contributed by cargo keeps increasing for the last decade, the cargo flow is very important for combination carriers and should also be properly modeled. The route of cargo is determined to a large extent by the cargo capacity of every leg, which depends on the fleet assignment decision. As a result, the fleet assignment has great influence on the cargo revenue. A traditional approach to the fleet assignment may cause great loss in the cargo revenue and thus the total profit of an airline. Incorporating the cargo routing into the fleet assignment can help the combination carrier to better balance its resource and the forecasted cargo demand. Motivated by this reason, we propose in this thesis an integrated approach to simultaneously determine the assignment of fleet to each leg and the cargo routing over the flight network.

Given a non-fleeted flight schedule that defines when and where to fly, a set of various aircraft, and forecasted cargo demands in all markets, we introduce an integrated model that combines the fleet assignment and the cargo routing problems. The cargo routing problem is to transport goods by aircraft from different origins to different destinations over a given network, with maximum revenue and without exceeding either aircraft capacity or a designated time window. Unlike the passenger, cargo has no strong preference on the specific itinerary as long as it can be delivered on time. This freedom increases the complexity of the problem. There is also no available industry data to calculate the spill cost and the recapture rate for the cargo flow. Furthermore, generally cargo is allowed to transfer between different aircraft only at the hub, while this requirement is not applicable to the passenger who is indifferent about whether the transit station is the hub or a spoke. As a result, the cargo flow problem is modeled in a much different way from the passenger flow problem.

1.4 Research Contributions

The main contributions in this thesis are:

- We present an integrated model of the fleet assignment with the cargo routing and develop a solution approach, referred as the basic algorithm, based on Benders decomposition. A series of computational experiments are carried out for different data sets to test its performance. Results show that the basic algorithm converges very fast for all test instances.
- We implement two different techniques to explore the possibility of accelerating the convergence of the basic algorithm. The pareto-optimal cut generation

approach solves an auxiliary linear programming problem at each Benders iteration to select a strong cut. To alleviate the burden of solving a large number of integer programs, the ε-optimal approach is designed to solve the Benders relaxed master problem to a feasible solution rather than an integer optimum. The objective function of the relaxed master problem is modified dynamically in each iteration so as to obtain a good feasible integer solution.

- We suggest a hybrid approach to enhance the ε-optimal approach. This solution approach first solves the original problem to a good feasible solution by the ε-optimal approach. After that, it turns to the basic algorithm to obtain the true optimal solution or a solution with a smaller optimality tolerance.

1.5 Organization of This Thesis

The rest of this thesis is divided into 5 parts. In Chapter 2, a literature review on the fleet assignment and the air cargo transportation is presented, followed by a thorough survey of the integrated airline schedule planning, where we can find different integrations of optimization problems in the airline industry developed so far.

Chapter 3 introduces all the mathematical formulations in our problem. A time-space dynamic network is constructed firstly, on the basis of which the basic fleet assignment model is built. After defining the commodity in our problem, we apply a two phase modeling technique to capture the time window and complex operational rules, and model the cargo routing problem as a path-oriented multicommodity network flow problem. The integrated formulation is finally presented as the combination of these two problems, with some modifications on the variables and constraints in the cargo routing model.

After a short review on Benders decomposition, Chapter 4 introduces the Benders reformulation of the integrated model. Then three different solution approaches are developed based on Benders decomposition and its two variants, Pareto-optimality cut generation and the ε-optimal approach, respectively. Both the solution methodologies and the specific solution procedures are described.

In Chapter 5, we describe our test instances and report the computational results. An improved solution approach that combines the basic algorithm and the ε-optimal approach is suggested. Finally a comparison of performances between different approaches is presented.

Chapter 6 summarizes the main results obtained from this research project, and discusses possible directions for future research in the integrated airline schedule planning.

2 Literature Review

We first outline previous related works in the fleet assignment problem in Section 2.1 and the air cargo shipment delivery problem in Section 2.2. In Section 2.3 we present a detailed review on the integrated airline planning.

2.1 Fleet Assignment

The fleet assignment problem is of considerable importance to airlines. Much research has been done to solve the daily fleet assignment problem, which determines a fleeting that remains the same every day of a week.

One of the early works is from Abara (1989), who built an integer linear programming model to solve the fleet assignment problem, with the objective to maximize the revenue of the flights minus the cost of the aircraft shortage and the cost of the stations. The model can be used either in the case where all legs are to be served or in the early stage of schedule planning when some legs may be dropped. The decision variables were defined for each feasible turn and aircraft combination. A *feasible turn* is a connection between two legs whose transit time is greater than a designated minimum turn time. There were also the balance or shortage variables and extra aircraft variables. Four out of the five main sets of constraints were flight coverage, continuity of equipment, schedule balance, and aircraft count, respectively. The fifth constraints can be lower bounds and upper bounds on any flight related variables, such as the limits on the overnight aircraft for a group of stations.

In 1995, Hane et al. modeled the daily fleet assignment as a large multicommodity flow problem and proposed efficient solution approaches to solve it. The formulation was defined on a set of time-space networks, each of which was constructed for a fleet type. Different from the work by Abara (1989), this paper defined two sets of decision variables. The binary fleet assignment variable was defined for each leg and fleet type combination, while the continuous ground arc variable was defined for each ground arc in the networks. To reduce the problem size, some adjacent nodes along a time line were aggregated and islands were constructed in spokes where the daily activity was very sparse. An *island* is an interval of time where there exists at least one aircraft on the ground. After testing different algorithms to solve the LP relaxation at branch and bound nodes, the authors concluded that the predictor-corrector interior point algorithm was much faster than the simplex method, but the time to go from an optimal non-basic solution to an optimal basic solution, the *crossover*, was even greater than the time to find the optimal solution in some cases. To avoid degeneracy in the problem, the cost coefficients of the ground arcs were perturbed after finding an optimal interior solution and before crossover. They also studied the branching rule in order to enhance the branch and bound. Instead of branching on a single variable, they suggested to branch on a cover row (flight cover constraint that was defined for every leg) so as to give a more balanced tree. To determine which cover row to branch at a node, each leg was issued a priority order that was obtained based on the variance in the cost coefficients of the variables covering that leg. Furthermore, if several branches had the same priority, a criterion was established to find the one that had the least harm to the objective function.

Subramanian et al. (1994) presented a fleet assignment model at the Delta airline. Other than the intrinsic constraints, the model also takes into account a lot of operational restrictions, including aircraft maintenance, pilot training, pilot hours, crew breakout, crew rest, and noises. Although some of the restrictions were modeled by the constraints, the others were coded into the input data.

A new fleet assignment model defined on an event-activity network was introduced by Rushmeier and Kontogiorgis (1997). Each node in this network was an *event* that represented the start of an operation (i, e), a flight i operated by fleet e. Each *flight activity arc* (i, j, e) represented a feasible connection between leg i and leg j by the same fleet type e. There was also a set of *sit activity arc*. The constructed fleet assignment model was able to capture connection rules of arbitrary complexities. Furthermore, it modeled constraints of resources, such as the aircraft maintenance and crew, as piecewise linear penalty terms, which allowed for a profit-maximization-based tradeoff between operational goals and revenue.

After the fleet assignment problem is solved, planners spend significant efforts in fine tuning and modifying the obtained solution to reflect business judgment calls that can not be captured by the optimization model. This process is referred as re-fleeting. Jarrah et al. (2000) formulated the re-fleeting problem as a multicommodity integer network flow problem, which comprehensively considered the practical re-fleeting questions. Used together with a fleet assignment model, the re-fleeting model can produce high-quality fleet assignment solutions. Another work in this area is from Talluri (1996), who introduced a simple algorithm to swap the fleet type of some legs without violating any constraint.

Berge and Hopperstad (1993) proposed models and algorithms for demand driven dispatch, which was an operational concept of fleet assignment. Taking advantage of improved demand forecast as flight departures approach, aircraft were assigned to legs dynamically to better match the predicted final demand. Demand forecasting was provided by the yield management system. Implementation of this concept required frequent solutions of the fleet assignment problem.

2.2 Air Cargo Shipment Delivery

The air cargo shipment delivery problem is seldom studied individually, but rather embedded into the service network design. Kim et al. (1999) modeled and solved a large scale service network design problem involving the express package delivery. After the service network was determined, the flow problem of package over the obtained service network was solved. The package flow problem was modeled as a multicommodity network flow problem in the tree formulation, which had one variable for each tree commodity. A commodity was defined by a single origin and possibly, multiple destinations. The objective was to move packages from their origin to their destinations with the minimum cost, satisfying service commitments and network capacity. Compared with the path-oriented formulation, the tree formulation reduced the number of constraints, because it had a smaller number of commodities and therefore a smaller number of demand constraints. However, the number of variables was increased exponentially relative to the number in the path formulation. Column generation was used to solve the tree formulation. Computational results showed that the tree-oriented formulation was solved much faster than the path-oriented formulation, which can be attributed to the relative uncongested service network. Other works in this area are Barnhart and Schneur (1996), Grainic (2000), and Barnhart et al. (2002).

Antes et al. (1998) developed a Decision- Support System (DSS) to evaluate the flight schedule for cargo airlines. Given the estimated market demand, a flight schedule, and transportation and handling capacities, their models found the maximum profit flow of cargo over the network. The system allowed different analysis on the flight schedule and the market share acquisition, which were quite useful for the tactical planning of the cargo airline. Two underlying models were built to enable the analysis. One was the basic multicommodity network flow model and another was the flexible model that included the time constraints.

2.3 Integrated Airline Schedule Planning

Few references can be found in the operations research literature regarding to the integration of optimization problems in the airline industry. The related previous works are classified into different categories according to the problems the integrated model involves.

2.3.1 Fleet Assignment with Aircraft Routing

One of the early efforts in this direction is the work of Barnhart et al. (1998a), who proposed a string-based integrated model to simultaneously solve the fleet assignment and the aircraft routing problems. A string was defined as a sequence of connected legs beginning and ending at the same maintenance station, without violating the flow balance and the maintenance feasibility requirements. This model can capture aircraft connection costs and complicating constraints such as maintenance requirements or aircraft utilization restrictions. However, the flight schedule with hundreds of legs had millions of strings, which resulted in a huge number of variables in the model. To overcome this drawback, this integrated model was solved by a branch-and-price algorithm, which is developed on the basis of branch-and-bound, but the LP relaxation at each node is solved by column generation.

Ioachim et al. (1999) introduced a fleet assignment and aircraft routing formulation with schedule synchronization constraints, which was solved by an approach based on Dantzig-Wolfe decomposition. The master problem contained flight covering constraints and schedule synchronization constraints. The resulting subproblem was a shortest path problem with time windows and was solved by a dynamic programming algorithm. A special branching strategy on the time variables was devised to find an optimal integer solution.

2.3.2 Fleet Assignment with Crew Scheduling

Another work of Barnhart et al. (1998b) is the integrated approximate modeling of the fleet assignment and crew pairing problems. Since the crew pairing problem takes very long time to solve, an approximate duty based model (Barnhart and Shenoi, 1998c) was used in replacement of it to maintain the solvability. This model was defined on a duty network, which had ground arcs and duty arcs (rather than flight arcs). Each duty arc represented a duty period. The integrated fleet assignment and crew pairing problem was solved by an advanced sequential solution approach. The approximate integrated model was solved first. Given its fleeting solution, an exact crew pairing problem was optimized for each fleet family. Therefore, it was not really a simultaneous optimization and the solution was not necessarily optimal.

Clarke et al. (1996) incorporated the maintenance and crew considerations into fleet assignment so as to generate feasible or improved solutions. To accomplish it, they added to the basic fleet assignment model the long and short maintenance constraints. To avoid lonely overnight for crews, a cost was issued to each lonely overnight and the objective of the optimization model was designed to balance the costs between lonely overnight and fleeting. This model improved the quality of the solution and can be solved with a reasonable increase in computer time.

2.3.3 Fleet Assignment with Schedule Development

Allowing scheduled flight departure time to change within a time window may improve flight connection opportunities and therefore generate a more profitable fleet assignment.

Rexing et al. (2000) presented a generalized fleet assignment model to simultaneously assign aircraft to legs and schedule the departure times. To model time windows, the basic flight time space network was extended by placing copies of a flight arc at specified intervals and requiring just one of them to be flown. This model differed from the basic fleet assignment model in that it had more variables and slightly different coverage constraints. To exclude many unnecessary flight copies from the model, an iterative solution technique was used to solve the problem. This approach first solved a fleet assignment problem with one arc for each flight and fleet combination. Additional flight copies were added iteratively if permitting a flexible departure time to these flights may improve the solution. By adding flight copies only when necessary, the iterative approach minimized the problem size and memory usage.

A similar work is from Desaulniers et al. (1997), who presented an integration of the fleet assignment model and time windows. Two equivalent models were constructed, a set partitioning formulation with side constraints and a time constrained multicommodity network flow formulation. Column generation and a Dantzig-Wolfe decomposition based approach were employed to solve the problem.

2.3.4 Fleet Assignment with Passenger Flow

Particularly interesting is an integrated model of the fleet assignment problem and the passenger mix problem introduced by Barnhart et al. (2002) to generate improved solutions. Given a fleeted schedule and unconstrained itinerary demand, the passenger mix problem optimizes the flow of passengers over this schedule with maximized revenue or minimized assignment cost. This itinerary-based fleet assignment model was able to capture network effects and estimate spill and recapture of passenger more accurately. In the integrated model, spill, recapture and their associated costs were decisions, which were constrained by the capacity assigned to the network. A restricted master problem (RMP) without passenger demand constraints and a subset of spill variables was constructed and the LP relaxation of RMP was solved by column and row generation. A heuristic IP solution approach was then employed to obtain the integrality solution.

2.3.5 Aircraft Routing with Crew Scheduling

Recently, Cordeau et al. (2001) introduced an integrated model and a solution approach based on Benders decomposition to optimize the crew pairing and the aircraft routing at the same time. The linear relaxation of the combined model was decomposed into a master problem that solves the aircraft routing problem in addition to a set of Benders cuts and a subproblem that solves the crew pairing problem. To avoid generating feasibility cuts from extreme rays, artificial variables with large positive costs were introduced to eliminate the infeasibility of the subproblem. At each Benders iteration, the special structure of the relaxed master problem and the subproblem enabled them to be solved by column generation. Optimal integer solution of the integrated model was obtained by a three phase approach.

Other contributions to improve the crew pairing solution by incorporating other planning problems are the works of Klabjan et al (2002) and Cohn and Barnhart (2003). Klabjan et al (2002) proposed to integrate the crew pairing partially with the schedule planning by allowing the shift of the departure time of every flight within a time window and the aircraft routing by adding plane-count constraints to the basic crew pairing model. This approach provided significant cost saving to the crew pairing problem compared to the basic model. Cohn and Barnhart (2003) presented an extended crew pairing model that was built upon the basic crew pairing formulation in addition to a collection of variables, each of which represented a complete solution to the maintenance routing problem. All the constraints in the original maintenance routing problem were therefore replaced by a single convexity constraint. However, the number of variables increased drastically because the number of feasible maintenance routing solutions was an exponential function of the number of legs. Realizing the fact that only part of the maintenance decisions affected the crew pairing problem, the authors reduced the problem size by including only one column representing a *unique* and *maximal* maintenance feasible short connect set. The resulting model had the flexibility to include additional airline planning decisions.

3 Mathematical Formulations

This chapter describes the underlying time-space dynamic flight network and the mathematical models of the fleet assignment problem (Section 3.1), the cargo routing problem (Section 3.2) and their combination (Section 3.3).

We base our models on the planning problems faced by a major Asia-pacific combination carrier. It operates a weekly flight schedule through a passenger network with six different fleets and a freighter network with only one fleet. The entire network has only one hub.

3.1 Fleet Assignment Model

Given a flight schedule and a set of fleets, the fleet assignment is made to determine which fleet to assign to each leg with the objective to minimize the total assignment cost or to maximize the total fleet assignment contribution. The fleet assignment problem is normally modeled as a multicommodity flow problem defined on a time-space dynamic flight network. Since the airline we are interested in operates a weekly schedule, the network covers all flights in a week and the model constructed is a weekly fleet assignment model.

The multicommodity network flow problem (MCNF) has been studied extensively. Ahuja et al. (1993) presented a thorough description of MCNF. Jones et al. (1993) studied the impact of the formulations on decomposition solution approaches for the multicommodity network flow problem. Three formulations, the node-arc, the path and the tree formulations were compared. The result showed the second formulation outperforms the other two in general cases. In our problem, both the fleet assignment and the cargo routing problems are modeled as MCNFs, but they are in different forms. Unlike the node-arc oriented fleet assignment model, the cargo routing model we build is path-oriented.

3.1.1 Time-Space Dynamic Flight Network

The weekly flight network is constructed in the same manner as the daily network. It contains leg arcs corresponding to legs and ground arcs corresponding to aircraft on the ground.

Let L be the set of legs in the weekly flight schedule indexed by i, S the set of stations served by the schedule indexed by s and E the set of fleet types indexed by e whose size is denoted by Num^e. To force the aircraft to circulate through the network, a time line for each station s and fleet e pair is constructed to model weekly activities. Along a time line the departure and arrival events in a week are represented as nodes in order of event time, which is defined as the departure time for departures and the arrival time plus the turn time (or ready time) for arrivals. The turn time is used for refueling, cleaning, package handling, etc. It varies mainly with fleet types and stations, and ranges from 30 to 60 minutes. A node n, $n \in N$ on a time line is denoted by (e, s, t), where t is the event time of this node. Let T denote the set of event times at a station, indexed by t_i. The event at t_i occurs before the event at t_{i+1}. If $|T|=m$, t_m is the last event before the count time, and t_1 is the first event after the count time. The count time is a specified time point in a week used to count the number of aircraft in the schedule. To facilitate the counting, the count time is best chosen at a time when most of the aircraft are on the ground. Connecting time lines at all stations by leg arcs forms a time-space network for each fleet. For every leg *i*, the pair of nodes created at its departure station and its arrival station are linked by a leg arc representing the assignment of a fleet to that leg. The node (e, s, t_n) along the time line at station *s* for fleet *e* is connected to its immediate successor node (e, s, t_{n+1}) by a ground arc. Since the schedule repeats itself weekly, the end of a time line is connected to the beginning by a cross week ground arc. Let *I(n)* be the set of inbound legs to node *n* and *O(n)* the set of outbound legs from node *n*. The set of legs that pass the count time and is flown by fleet type *e* is denoted by *O(e)*. A station and fleet dependent time line is shown in Figure 3.1, from which we can clearly find all feasible connections between legs.

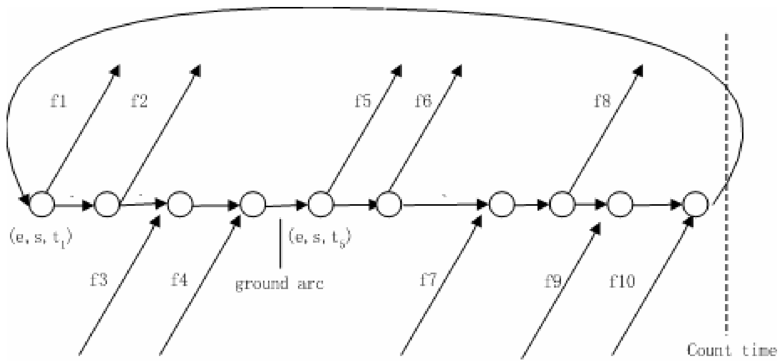

Figure 3.1 A Station-Fleet Time Line

Because the turn time varies with the fleet type, we should construct a special network for each fleet. In this research, however, we assume the turn times are the same for all fleets at all times at each station, thus only one time space network is needed for all various fleet types. Such an assumption simplifies the modeling process without compromising the research objective, which is to explore modeling and solution approaches to the integrated problem.

3.1.2 Basic Fleet Assignment Model

Prior to the mathematical formulation of the basic fleet assignment model, the following notations are defined, some of which have been used in the previous Section 3.1.1.

Sets

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

E: the set of different fleet types indexed by e.

S: the set of stations indexed by s.

T: the set of arrival and departure event times at all stations in a week, indexed by t_i.

N: the set of nodes in the time space network indexed by n or (e, s, t), where $e \in E, s \in S, t \in T$.

$O(e)$: the set of legs that pass the count time and is flown by fleet type e.

$I(n)$: the set of inbound legs to node n or (e, s, t).

$O(n)$: the set of outbound legs from node n or (e, s, t).

Parameters

Num^e: the number of aircraft in fleet type e.

c_i^e: the cost of assigning fleet type e to leg i, which is the operational fleet assignment cost minus the estimated passenger revenue, written as $c_i^e = c_{(0)_i}^{\ e} - \tau_i^e$. The coefficient $c_{(0)_i}^{\ e}$ is the operational cost that includes the cost of fuel, handling, take-off and landing. τ_i^e is the estimated passenger revenue of flying leg *i* with fleet *e*, which is a linear function of the seat number and the block time of this leg. The revenue per seat per time unit is obtained from the annual financial report of the airline. We ignore the spill cost and the recapture but simply estimate the passenger revenue linearly, for this research project mainly focuses on the cargo flow segment.

Decision variables

$$x_i^e = \begin{cases} 1, \text{ if flight leg } i \text{ is assigned to fleet type } e; \\ 0, \text{otherwise} \end{cases}$$

$g_{e,s,(t_n,t_{n+1})}$: the number of fleet type *e* that are on the ground at station *s* immediately after event time t_n or just before event time t_{n+1}.

The mathematical formulation of the basic fleet assignment model (FAM) is:

$$\min \sum_{i \in L} \sum_{e \in E} c_i^e x_i^e \tag{3.0}$$

subject to:

$$\sum_e x_i^e = 1, \forall i \in L \tag{3.1}$$

$$g_{e,s,(t_{n-1},t_n)} + \sum_{i \in I(n)} x_i^e = g_{e,s,(t_n,t_{n+1})} + \sum_{i \in O(n)} x_i^e, \forall n \in N, e \in E \tag{3.2}$$

$$\sum_{s} g_{e,s,(t_m,t_1)} + \sum_{i \in O(e)} x_i^e \le Num^e, \forall e \in E \tag{3.3}$$

$$x_i^e \in \{0,1\}, \forall i \in L, e \in E$$
$$g_{e,s,(t_n,t_{n+1})} \ge 0, \forall e \in E, s \in S, t_n, t_{n+1} \in T$$

The objective (3.0) minimizes the total fleet assignment costs or maximizes the total passenger revenue minus the operational costs. Cover constraints (3.1) ensure that each leg is covered once and only once by a fleet type. Balance constraints (3.2) ensure that aircraft going into a station at a particular time must leave the same station at some time later. The final set of constraints (3.3) ensures that the number of every fleet in use does not exceed the total number available of this fleet.

Because of the large number of nodes in the network, this model has much more balance constraints ($|N||E|$) than the other two constraints $(|L|+|E|)$. Similarly, the number of ground arc variables ($|N||E|$) is much more than that of the binary variables ($|L||E|$). The problems size, therefore, may reduce greatly with the reduction of the number of nodes. To accomplish it, Hane et al. (1995) proposed an idea of node aggregation. Since the model does not care about the specific arrival (departure) time as long as the correct connection opportunity is guaranteed, there is no use to create arrival (departure) nodes in the time line earlier (later) than the next departure (previous arrival). In other words, only one node is created in the time line for consecutive arrivals followed by consecutive departures. This node spans a time interval which begins with the first arrival and ends with the last departure. After node aggregation, the time line in Figure 3.1 has much fewer nodes, as shown in Figure 3.2.

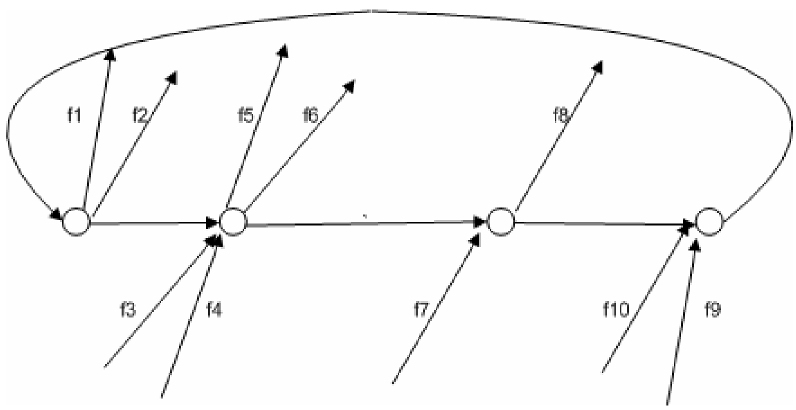

Figure 3.2 The Station-Fleet Time Line After Node Aggregation

Aggregating nodes will greatly reduce the number of balance constraints and the ground arc variables. A further reduction of the problem size can be achieved by constructing islands (Hane et al. 1995) in spokes where the daily activity is very sparse. Since there is no aircraft on the ground between two islands, the ground arc connecting them is removed. Figure 3.3 shows the time line after removing three ground arcs. Particularly, for an island with only one arrival followed by one departure, like the third node in Figure 3.3, the corresponding pair of fleet assignment variables are combined into one variable that represents the assignment to the path or the pair of legs. The path can be extended to include three or more consecutive legs. As a result, the number of binary decision variables is reduced.

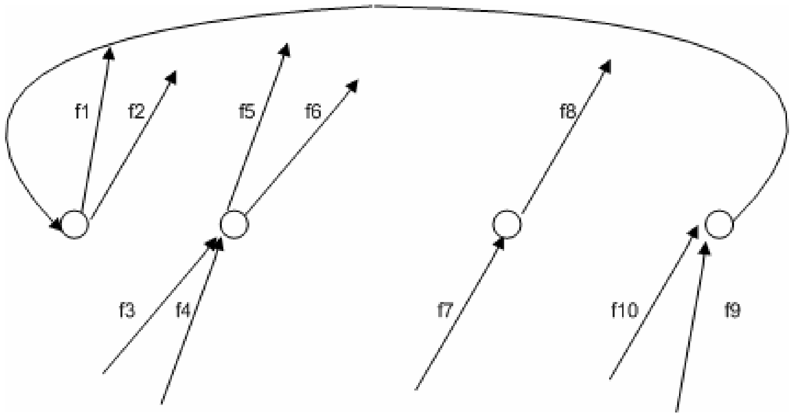

Figure 3.3 The Station-Fleet Time Line
After Node Aggregation And Removal Of Ground Arcs

These two problem size reduction methods only modify the representation of the constraints and the decision variables. We still use the formulations (3.0)-(3.3) to represent the fleet assignment model in our later work.

3.2 Cargo Routing Model

Given the fleeted flight schedule and the forecasted unconstrained cargo demand in all markets, cargo routing maximizes the revenue of the commodity flow without exceeding the flight capacity and the time window. Three additional side constraints should also be satisfied. Firstly, the carrier allows cargo to be transferred from one aircraft to another only at the hub and at no other stations. Since a lot of work and time is needed to offload cargo from one aircraft and load on to another, it is desired to be done at the hub where the airline has sufficient cargo handling facilities and work forces to complete transferring works quickly and cheaply. Secondly, if transferring happens at the hub, the transit time between the two connected legs must be longer than the minimum cargo handling time. Finally, cargo is not allowed to backtrack between spokes.

In practice, the cargo network for some airlines includes not only the air network. They also have the trucking network, which is an important part of the entire cargo transportation system. Such trucking transportation is quite popular in European countries. For example, if a commodity needs to be shipped from Singapore to Copenhagen, it can be firstly transported to Frankfurt by air, and then to Copenhagen by trucking. The flow of air cargo will be influenced greatly by the truck capacities. Thus the trucking network should be included when optimizing cargo flow. However, there are many difficulties to model the entire cargo network. The trucking schedule is very hard to get; the demand becomes more complicated to forecast; the size of the problem increases drastically. In this thesis we mainly focus on the air network and leave the trucking network aside. Future research may include the trucking network if the necessary data are available.

3.2.1 Definition of the Commodity

Because we model cargo routing as a path-oriented multicommodity network flow problem, the commodity should be origin-destination specified. To capture the time requirement of the cargo delivery, time elements are added to the definition of the commodity.

A commodity in our problem is defined by (o, t, D_o, T_o, T_w), where:

o: the specific origin, $o \in S$.

t: the specific destination, $t \in S$.

D_o: the day on which the commodity is ready for shipment at the origin, from Monday to Sunday,

T_o: the ready time for shipment at the origin, either in the morning, at noon or in the evening.

T_w: the time window for the commodity, one day (24hrs) or two days (48hrs).

For example, one commodity in our problem could be the cargo that is available for shipment in Bangkok on Monday evening and should be transported to Tokyo within two days.

3.2.2 Feasible Path Criteria

A direct path for a commodity consists of a set of consecutive legs that start from the origin and end at the destination of this commodity. This path is feasible if the following requirements are satisfied.

1 The cargo can only be transferred to another aircraft at the hub. This means all legs in this path must use the same aircraft or tail number unless it transits at the hub. In that case, at most two tail numbers can be used, one for legs prior to the hub and one for legs after the hub.

2 If transferring happens, the connecting time between the two legs must be longer than the minimum cargo handling time at the hub.

3 The time length of this path must be within the time window specified for the commodity. The time length is the elapsed time from the ready time to the arrival time at the destination.

4 Cargo is not allowed to backtrack between spokes. If a path does not pass through the hub, the legs in it must fly roughly towards one direction, which is enforced by restricting the number of passenger/freighter legs in this path to

2/4. For a path transferring at the hub, the legs can change flying direction only when leaving the hub, which is enforced by restricting the number of passenger/freighter legs before (after) the hub to 2/4 in this path.

If a flight schedule segment in Table 1 is available and SIN is the hub, the above commodity example may have two feasible paths, as shown in Figure 3.4. If SIN is a spoke, the path *f1*-*f2* becomes infeasible, because two different tail numbers are assigned to these two legs.

Table 3.1 A Fight Schedule Segment

Leg	Origin	Departure Day	Departure Time	Destination	Arrival Time	Plane Tail #
f1	BKK	2	1440	SIN	1805	NA01
f2	SIN	2	2220	NRT	0545	NA02
f3	BKK	2	1205	NRT	1955	NA03

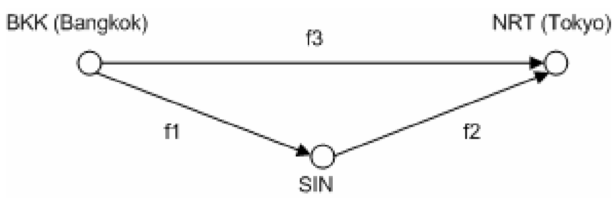

Figure 3.4 Feasible Path Examples

The feasible paths are generated based on the given fleeted air network, which will change with the fleet assignment decision. As we will address later, this characteristic adds complexity to the integrated model construction.

3.2.3 Modeling Approach

Cargo routing is typically modeled as a multicommodity network flow problem, but explicitly modeling the time window constraints and the side constraints will result in a very complicated model. To capture these complex rules while maintaining the tractability, we apply a two phase modeling approach.

1. All feasible paths are generated for all commodities.
2. A MCNF path formulation with only the columns of feasible paths and rows of capacity and demand constraints is built.

Compared with explicitly modeling every constraint in the path formulation, this approach not only simplifies the cargo routing model by removing the difficult constraints, but also greatly reduces the number of columns by excluding infeasible paths. The resulting model can be solved directly by the simplex algorithm.

3.2.4 Path-Oriented Cargo Routing Model

The following notations are defined for the cargo routing model.

K: set of commodities indexed by k.

L: set of legs in flight schedule indexed by i.

$P_f(k)$: set of feasible paths for commodity k, indexed by p.

d_i: cargo capacity of leg i, which depends on the fleet assigned to it.

B^k: unconstrained demand of commodity k.

r_p^k: per unit selling price or revenue of flowing commodity k on path p;

$$\delta_i^p: \quad = \begin{cases} 1, \text{ } if\ i \in p \\ 0, \text{ } otherwise \end{cases}$$

Decision variable:

y_p^k: the amount of commodity k flown on path p.

The mathematical formulation of the cargo routing model (CRM) is:

$$\max \sum_{k \in K} \sum_{p \in P_f(k)} r_p^k y_p^k \tag{3.4}$$

subject to:

$$\sum_{k \in K} \sum_{p \in P_f(k)} y_p^k \delta_i^p \leq d_i, \forall i \in L, \tag{3.5}$$

$$\sum_{p \in P_f(k)} y_p^k \leq B^k, \forall k \in K \tag{3.6}$$

$$y_p^k \geq 0, \forall p \in P_f(k), k \in K \tag{3.7}$$

The objective of CRM (3.4) maximizes the total cargo revenue. The capacity constraints (3.5) restrict the total cargo flown on a leg to its cargo capacity. By flow constraints (3.6), the cargo actually shipped is less than or equal to the unconstrained demand.

The coefficient r_p^k is determined in the following way.

1. For every commodity, there is a unit rate ρ^k, which is dependent on the characteristics of the commodity, specifically, the origin-destination distance and the time window. The longer the distance, the higher the rate. The shorter the time

window, the higher the rate. We ignore the discount for large demand and assume the unit rate is the same for whatever demand quantities.

2. There may be different feasible paths with different elapsed time for a commodity. We introduce a pseudo cost c_p^k to account for such a difference and to give incentives for the optimizer to choose the shorter path. This can be translated to improving the service level. That is, the service level will be improved if the commodity is shipped on shorter paths. The pseudo cost c_p^k for commodity k on path p is computed by (T/T_w)*0.001, where T is the elapsed time of the path, and T_w is the time window of the commodity.
3. For commodity k on path p, the per unit revenue $r_p^k = \rho^k - c_p^k$

3.3 Integrated model

The objective of the integrated model is to maximize the estimated total cargo and passenger profit by incorporating the cargo routing into the fleet assignment. Usually the network of a combination carrier can be divided into two subnetworks according to the fleet type, one for the passenger fleet flow and the other for the cargo fleet flow. Since the freighter can not be assigned to passenger flights, and vice visa, the fleet assignment should be done separately for these two networks. However, cargo routing must take account of the capacity available on both simultaneously, because cargo can also flow on passenger flights, in the belly of the passenger aircraft. The feasible paths are thus constructed over the entire network. The conceptual general integrated fleet assignment and cargo routing model for multiple passenger fleet types and multiple freighter types is shown in Figure 3.5. The passenger FAM and the freighter FAM are independent of each other, but they interact with the CRM by the capacity constraints and the feasible paths.

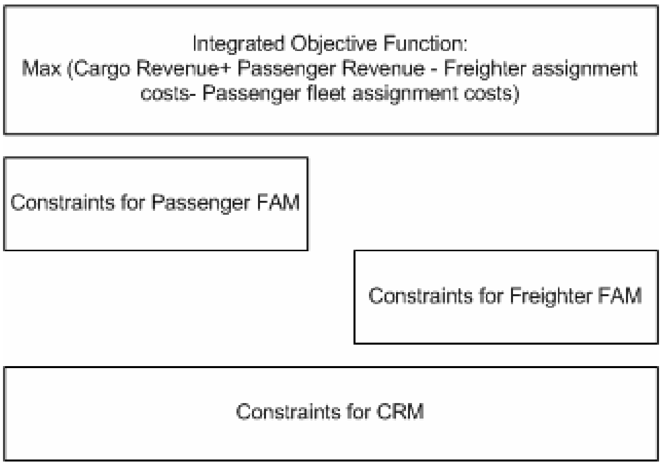

Figure 3.5 Conceptual Integrated Model

To obtain the integrated model we combine the passenger FAM, the freighter FAM and CRM together and link them by multiplying the capacity constant in the CRM capacity constraints by the fleet assignment variable. As the combination carrier in question has only one type of freighter, the fleet assignment is not required for the freighter flights. The integrated model thus includes only the passenger FAM and the CRM.

3.3.1 Justification of the Integrated Model

This integrated formulation comprises the fleet assignment model and the cargo routing model, but ignores the aircraft routing problem. We know that one side constraint for the cargo routing problem requires cargo can be transferred to other aircraft only at the hub. Therefore, an integrated model should be able to identify different physical aircraft or tail number by incorporating the aircraft routing, which leads to a three stage problem. The first stage would deal with the fleet assignment, the second stage the aircraft routing, and the third stage the cargo routing. Nevertheless, we can justify our two stage integrated model approximates the actual three stage problem very well for the carrier in question. For most spoke stations, the frequency of legs is quite low. The typical activity at a spoke consists of an arrival followed by a departure. For instance, the airline flies 98 legs a week to and from a busy station. The first leg departs at 0655 in Monday morning, followed by 48 pairs of inbound and outbound legs. The last leg arrives at 2150 in Sunday evening. Figure 3.6 depicts the time line in this station after removing the ground arc.

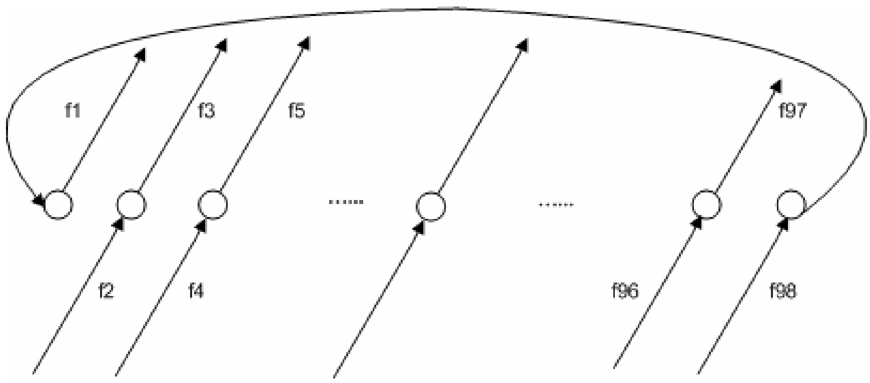

Figure 3.6 Time line at A Spoke

To minimize the number of aircraft used at this station and to avoid having unnecessary aircraft on the ground overnight, the aircraft flying the arrival flight is usually assigned to the immediate departure flight. At spokes islands are constructed and most of them have only one arrival followed by one departure. As a result, the same fleet types for the two connecting legs of an island will be reasonably taken to imply the same physical aircraft. Although this is not the case for the hub, we do not care about the physical aircraft because cargo can transfer at the hub. This observation justifies that our two stage integrated model is a reasonable approximation to the actual problem, which could be too complicated to be solved efficiently.

3.3.2 Model Dynamic Feasible Paths

Since the fleet assignment and the cargo routing are determined simultaneously, the feasible paths are not fixed but altered whenever the fleet assignment changes. To accommodate this uncertainty, all potential feasible paths must be included in the integrated model. A direct path from the origin to the destination of a commodity is regarded as potentially feasible if it meets the last three criteria of a feasible path. Let $P(k)$ denote the set of potential feasible paths for commodity k. The potential feasible path is generated based on the non-fleeted flight schedule and becomes feasible as soon as its any two constituent legs connected at a spoke are assigned with the same fleet.

The dynamic character of the feasible path is captured by disaggregating the cargo capacity constraints and the flow decision variables. There is one capacity constraint for each leg and each fleet type combination. To define the disaggregated variables, we divide the set $P(k)$ into two subsets. The set $P^T(k)$ contains the paths transferring at the hub, and the other set $P^D(k)$ contains the rest of the paths. Since every path p in the set $P^T(k)$ may be assigned with two different fleet types, we split it into two *subpaths* (as shown in Figure 3.7), $i(p)$ arriving at the hub and $o(p)$ departing the hub, each of which can be assigned with only one fleet type. Accordingly, the set $P^T(k)$ is divided into $PI^T(k)$ and $PO^T(k)$, which contain all $i(p)$ and $o(p)$ for commodity k, respectively.

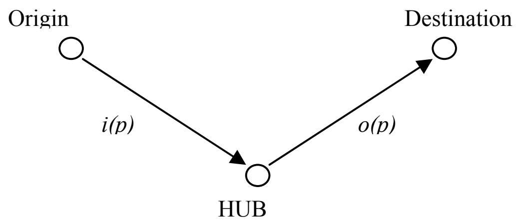

Figure 3.7 A transferring Path

All the paths in $P^D(k)$ are further classified into two groups by their constituent legs. Passenger path group $P_{pax}^D(k)$ contains paths flowing through only passenger legs, and freighter path group $P_{frt}^D(k)$ contains paths flowing through only freighter legs. Similarly, sets $PI^T(k)$ and $PO^T(k)$ are divided into $PI_{pax}^T(k), PI_{frt}^T(k)$ and $PO_{pax}^T(k), PO_{frt}^T(k)$, respectively. The disaggregated flow variables are then defined for every path or subpath and fleet combination.

To understand how the disaggregation solves the issue of the dynamic feasible path, let us take an example of a direct passenger path p_0, which consists of two legs i_1 and i_2. The disaggregated flow variables are then $y_{p0}^{k,e}, \forall e \in E(pax)$ and the disaggregated capacity constraints for this path are:

$$y_{p0}^{k,e} \le x_{i1}^e d^e, \forall e \in E(pax)$$
$$y_{p0}^{k,e} \le x_{i2}^e d^e, \forall e \in E(pax)$$

These constraints will force the two legs in this path to be assigned with the same fleet. Otherwise the variables $y_{p0}^{k,e} = 0, \forall e \in E(pax)$, which means the path is infeasible and no commodity is allowed to be shipped through it. Therefore, no matter what the fleet assignment is, only the feasible paths are active and all infeasible ones are excluded from the cargo routing solution. The infeasible paths can not be detected by the following aggregated constraints.

$$y^k_{p0} \le \sum_e x^e_{i1} d^e$$
$$y^k_{p0} \le \sum_e x^e_{i2} d^e$$

As long as a fleet is assigned to each leg, the commodity can be shipped along this path.

3.3.3 Mathematical Formulation

The additional notations used in the integrated formulation are defined below.

Sets

L(pax): the set of passenger legs.

N(pax): the set of nodes in the passenger time-space network.

E(pax): the set of passenger fleets.

L(frt): the set of freighter legs.

f: the freighter.

Parameters

d^e: the cargo capacity of passenger fleet $e \in E(pax)$.

d^f: the cargo capacity of the freighter f.

$r^k_{i(P)}$ per unit revenue of flowing commodity k on subpath $i(p)$, where $p \in P^T(k)$. Since the revenue is generated by the path p and not its subpath $i(p)$, we have $r^k_{i(P)} = r^k_p$.

Decision variables

$y_p^{k,e}$: flow variable corresponding to the direct passenger path $p \in P_{pax}^D(k)$ and $e \in E(pax), k \in K$.

$y_p^{k,f}$: flow variable corresponding to the direct freighter path $p \in P_{frt}^D(k)$ and $k \in K$.

$y_{i(p)}^{k,e}$, $y_{o(p)}^{k,e}$: flow variable corresponding to the passenger subpath $i(p) \in PI_{pax}^T(k), o(p) \in PO_{pax}^T(k)$, where p is a transferring path $p \in P^T(k)$ and $e \in E(pax), k \in K$.

$y_{i(p)}^{k,f}$, $y_{o(p)}^{k,f}$: flow variable corresponding to the freighter subpath $i(p) \in PI_{frt}^T(k), o(p) \in PO_{frt}^T(k)$, where p is a transferring path $p \in P^T(k)$ and $k \in K$.

The integrated formulation is:

$$\begin{aligned} \max \sum_{k \in K} (\sum_{p \in P_{pax}^D(k)} \sum_{e \in E(pax)} r_p^k y_p^{k,e} + \sum_{\substack{i(p) \in PI_{pax}^T(k), \\ p \in P^T(k)}} \sum_{e \in E(pax)} r_{i(p)}^k y_{i(p)}^{k,e}) \\ + \sum_{k \in K} (\sum_{p \in P_{frt}^D(k)} r_p^k y_p^{k,f} + \sum_{\substack{i(p) \in PI_{frt}^T(k) \\ p \in P^T(k)}} r_{i(p)}^k y_{i(p)}^{k,f}) - \sum_{i \in L(pax)} \sum_{e \in E(pax)} c_i^e x_i^e \end{aligned} \tag{3.8}$$

subject to:

$$\sum_{e \in E(pax)} x_i^e = 1, \forall i \in L(pax), \tag{3.9}$$

$$g_{e,s,(t_{n-1},t_n)} + \sum_{i \in I(n)} x_i^e = g_{e,s,(t_n,t_{n+1})} + \sum_{i \in O(n)} x_i^e, \forall n \in N(pax), e \in E(pax) \tag{3.10}$$

$$\sum_s g_{e,s,(t_m,t_1)} + \sum_{i \in O(e)} x_i^e \le Num^e, \forall e \in E(pax) \tag{3.11}$$

$$\sum_{k\in K}\sum_{p\in P^D_{pax}(k)} y_p^{k,e}\delta_i^p + \sum_{k\in K, i(p)\in PI^T_{pax}(k)}\sum_{p\in P^T(k)} y_{i(p)}^{k,e}\delta_i^{i(p)} + \sum_{k\in K, o(p)\in PO^T_{pax}(k)}\sum_{p\in P^T(k)} y_{o(p)}^{k,e}\delta_i^{o(p)} \le x_i^e d^e, \forall i\in L(pax), e\in E(pax) \quad (3.12)$$

$$\sum_{k\in K}\sum_{p\in P^D_{frt}(k)} y_p^{k,f}\delta_i^p + \sum_{k\in K, i(p)\in PI^T_{frt}(k)}\sum_{p\in P^T(k)} y_{i(p)}^{k,f}\delta_i^{i(p)} + \sum_{k\in K\, o(p)\in PO^T_{frt}(k)}\sum_{p\in P^T(k)} y_{o(p)}^{k,f}\delta_i^{o(p)} \le d^f, \forall i\in L(frt) \quad (3.13)$$

$$\sum_{p\in P^D_{frt}(k)} y_p^{k,f} + \sum_{p\in P^D_{pax}(k)}\sum_{e\in E(pax)} y_p^{k,e} + \sum_{\substack{i(p)\in PI^T_{frt}(k)\\ p\in P^T(k)}} y_{i(p)}^{k,f} + \sum_{\substack{i(p)\in PI^T_{pax}(k)\\ p\in P^T(k)}}\sum_{e\in E(pax)} y_{i(p)}^{k,e} \le B^k, \forall k\in K \quad (3.14)$$

$$\left\{\begin{array}{l} \sum_{e\in E(pax)} y_{i(p)}^{k,e} = \sum_{e\in E(pax)} y_{o(p)}^{k,e}, \text{if } i(p)\in PI^T_{pax}(k) \text{ and } o(p)\in PO^T_{pax}(k) \\ \sum_{e\in E(pax)} y_{i(p)}^{k,e} = y_{o(p)}^{k,f}, \text{if } i(p)\in PI^T_{pax}(k) \text{ and } o(p)\in PO^T_{frt}(k) \\ y_{i(p)}^{k,f} = \sum_{e\in E(pax)} y_{o(p)}^{k,e}, \text{if } i(p)\in PI^T_{frt}(k) \text{ and } o(p)\in PO^T_{pax}(k) \\ y_{i(p)}^{k,f} = y_{o(p)}^{k,f}, \text{if } i(p)\in PI^T_{frt}(k) \text{ and } o(p)\in PO^T_{frt}(k) \end{array}\right\}, \forall p\in P^T(k) \quad (3.15)$$

$$x_i^e \in \{0,1\}, \forall i\in L(pax), e\in E(pax);$$
$$g_{e,s,(t_n,t_{n+1})} \ge 0, \forall e\in E(pax), s\in S, t_n, t_{n+1}\in T;$$
$$y_p^{k,e} \ge 0, \forall p\in P^D_{pax}(k), k\in K, e\in E(pax); y_p^{k,f} \ge 0, \forall p\in P^D_{frt}(k), k\in K;$$
$$y_{i(p)}^{k,e}, y_{o(p)}^{k,e} \ge 0, \forall i(p)\in PI^T_{pax}(k), o(p)\in PO^T_{pax}(k), p\in P^T(k), k\in K, e\in E(pax);$$
$$y_{i(p)}^{k,f}, y_{o(p)}^{k,f} \ge 0, \forall i(p)\in PI^T_{frt}(k), o(p)\in PO^T_{frt}(k), p\in P^T(k), k\in K.$$

The objective (3.8) maximizes the total cargo and passenger profit. Its first two parts are cargo revenue contributed by the passenger network and the freighter network, respectively. The third part is net passenger revenue by the passenger flights. Since the freighter legs' assignment costs are constant, they are excluded from the objective function. We include only variables for subpath *i(p)*, but not o(*p)*, in the objective function to avoid double counting. For the same reason constraints (3.14) include only flow variables for *i(p)*.

The first three sets of constraints (3.9)-(3.11) are the constraints of the passenger FAM and the rest are the constraints of the CRM. Disaggregated capacity constraints (3.12)-(3.13) are defined individually for the passenger network and the freighter network. The right hand sides of constraints (3.13) are constant because only one type of freighter is available. Constraints (3.14) are the demand constraints. By Constraints (3.15), the flow consistency along a transferring path is ensured. For every path $p \in P^T(k)$, one of the four constraints is defined, according to the type of its subpaths. For instance, if its subpath *i(p)* flows through the passenger network and *o(p)* flows through the freighter network, the second constraint of (3.15) is defined for this transferring path p.

In comparison with those in the individual CRM model, the demand and capacity constraints in the integrated formulation are much more complicated. Moreover, a set of additional constraints—transferring constraints are defined. This is because we have no knowledge about the fleet type of passenger legs. These complex constraints will reduce to the simple form once the fleet assignment is known. Such a feature is utilized in our solution approach where the cargo routing model is set up and solved after the fleet assignment problem.

4 Solution Methodology

The real-life applications of problem (3.8)-(3.15) are too large to be solved economically by general mixed integer programming codes. The largest application presented later has about 8,400 binary variables, 1,200,000 continuous variables and 150,000 constraints. Fortunately however, the integrated model naturally decomposes into two subproblems that are relatively easy to solve. For any feasible solution to constraints (3.9) – (3.11) that involves only fleet assignment variables, problem (3.8)-(3.15) reduces to a cargo routing problem involving only cargo flow variables. This observation motives the development of the solution approach based on Benders decomposition (Benders, 1962).

Section 4.1 presents a review of the Benders decomposition algorithm. Section 4.2 reformulates our integrated formulation following this algorithm. Then three solution approaches based on Benders decomposition and its two variants are developed in Section 4.3 to Section 4.5.

4.1 Review of Benders Decomposition

Consider the following mixed-integer problem.

$$\begin{aligned} &\min \quad z = cx + dy \\ &st. \quad Ax + Gy \geq b \\ &x \in Z^+, y \in R^+ \end{aligned} \tag{4.1}$$

In some applications, if complicating variables, such as x, are fixed, the resulting problem becomes a relative easy problem. The Benders decomposition method assigns trial values to these variables and finds the corresponding best solution. In the process either an optimal solution of the original problem is found or an infeasibility/unboundedness is detected.

Benders decomposition begins with reformulating (4.1) as a problem that includes only a subset of variables by projecting out the others. Fix x at $\bar{x}$, the problem becomes:

$$\begin{aligned} \min \quad & z(\bar{x}) = dy + c\bar{x} \\ st. \quad & Gy \geq b - A\bar{x} \\ & y \in R^{+} \end{aligned} \tag{4.2}$$

We eliminate y from constraints first. Since only the feasibility of the above problem needs to be considered, the objective function is left aside and replaced by a constant zero.

$$\begin{aligned} \min \quad & 0 \\ st. \quad & Gy \geq b - A\bar{x} \\ & y \in R^{+} \end{aligned} \tag{4.3}$$

Its dual is:

$$\begin{aligned} \max \quad & \nu(b - A\bar{x}) \\ st. \quad & \nu G \leq 0 \\ & \nu \in R^{+} \end{aligned} \tag{4.4}$$

The primal problem (4.3) is feasible if its dual objective value is less or equal to 0 for all ν , and the equality holds for at least one ν . (Note the dual is always feasible.) That is,

$$\nu(b - A\bar{x}) \leq 0 : \nu G \leq 0; \nu \in R^{+} \tag{4.5}$$

Let $\{\nu_j^*, j \in J\}$ be the collection of extreme rays of the cone $C = \{\nu G \leq 0, \nu \geq 0\}$. The condition (4.5) is equivalent to

$$\nu_j^*(b - Ax) \leq 0, \forall j \in J \tag{4.6}$$

The specific $\bar{x}$ is replaced by the general x variables because the extreme rays of C are independent of x. Condition (4.6) is then the sufficient condition for the primal to be feasible. Multiply the constraint in (4.3) by all extreme rays of $C = \{vG \le 0, v \ge 0\}$. After relaxing variable x to general values, we have

$$\begin{aligned} & v_j^* Gy \ge v_j^*(b - Ax) \\ & \Rightarrow 0 \ge v_j^*(b - Ax),\ \forall j \in J \end{aligned}.$$

Therefore (4.6) is also the necessary condition for the primal to be feasible. The constraints in the original problem (4.1) is equivalent to the condition (4.6) that contains only x variables. For this reason, (4.6) is called the Benders feasibility cut.

Next we eliminate y from the objective function. Problem (4.2) can be expressed as

$$\begin{aligned} & \min \quad z(\bar{x}) = \eta + c\bar{x} \\ & st. \quad \eta - dy \ge 0 \\ & Gy \ge b - A\bar{x} \\ & y \in R^+ \end{aligned} \qquad (4.7)$$

Its dual is

$$\begin{aligned} & \max \quad \mu(b - A\bar{x}) \\ & st. \quad \mu_0 = 1 \\ & \mu G - \mu_0 d \le 0 \\ & \mu, \mu_0 \ge 0 \end{aligned} \qquad \text{or equivalently,} \qquad \begin{aligned} & \max \mu(b - A\bar{x}) \\ & st. \quad \mu G - d \le 0 \\ & \mu \ge 0 \end{aligned} \qquad (4.8)$$

As $c\bar{x}$ is a constant, it is not included in the dual objective function. If the primal problem (4.7) is feasible for the given $\bar{x}$, the dual (4.8) is either infeasible or has an optimal solution. In the case of dual infeasibility, (4.7) will be unbounded ($-\infty$), and so is the original problem (4.1). If the dual has an optimal solution, according to the strong duality the primal must have the same optimal value as the dual. That is

$$\eta \geq \mu_i^*(b - Ax), \forall i \in I \tag{4.9}$$

where $\{\mu_i^*, i \in I\}$is the collection of the extreme points of $Q = \{\mu \in R^+, \mu G \leq d\}$ and the equality holds for at least one extreme point. Since the extreme point of Q does not depend on the value of x, $\bar{x}$ is relaxed in (4.9).

This condition is also necessary for the primal problem to be optimal. Multiply the constraints in (4.7) by $\{1, \mu_i^*\}$, we get

$$\begin{aligned}
&\eta + \mu_i^* G y \geq dy + \mu_i^*(b - A\bar{x}) \\
&\Rightarrow \eta + dy \geq dy + \mu_i^*(b - A\bar{x}) \\
&\Rightarrow \eta \geq \mu_i^*(b - A\bar{x}) \\
&\Rightarrow \eta \geq \mu_i^*(b - Ax)
\end{aligned}$$

Now we also eliminate variable y from the objective function. (4.9) is referred to as the Benders optimality cut.

Given the above results, the original problem (4.1) can be reformulated to include only x variables. The Benders reformulation is written as follows:

$$\begin{aligned}
&\min_x z = cx + \eta \\
&st. \;\; \eta \geq (b - Ax)\mu_i^*, \forall i \in I \\
&v_j^*(b - Ax \;\;) \leq 0, \forall j \in J \\
&x \in Z^+
\end{aligned} \tag{4.10}$$

Compared to the original problem, the reformulation (4.10) generally contains a huge number of constraints. However, only a subset of them is active in an optimal solution. Instead of enumerating all constraints explicitly, the decomposition algorithm generates them on the fly, that is, on an as needed basis.

The Benders decomposition algorithm proceeds as follows. At each iteration H the Benders cuts generated so far are added to the following Benders *relaxed master problem*.

$$\begin{aligned}
&\min_x z^H = cx + \eta \\
&st.\ \ \eta \ge (b - Ax)\mu_i^*, \forall i \in I^H \\
&\nu_j^*(b - Ax) \le 0, \forall j \in J^H \\
&x \in Z^+
\end{aligned} \tag{4.11}$$

The solution of the relaxed maser problem provides a new value of $\bar{x}^H$, based on which a *Benders subproblem* is set up, which contains only the variables y.

$$\begin{aligned}
&\min z = c\bar{x}^H + dy \\
&st.\ \ Gy \ge b - A\bar{x}^H \\
&y \in R^+
\end{aligned} \tag{4.12}$$

If the solution of the relaxed master problem is equal to the best subproblem solution, the optimality of the original problem is reached. Otherwise, new Benders cuts are generated from the dual solution of the subproblem. At this point the procedure repeats.

4.2 Benders Reformulation of the Integrated Formulation

Now we apply the Benders decomposition algorithm to reformulate our integrated model and decompose it into two problems. The relaxed master problem contains fleet assignment variables and the subproblem contains cargo routing variables.

For any given value $\bar{x}_i^e, \bar{g}_{e,s,(t_n,t_{n+1})} (i \in L(pax), e \in E(pax), s \in S, t_n, t_{n+1} \in T)$ satisfying passenger fleet assignment constraints (3.9)-(3.11), the integrated model reduces to the following Benders *primal subproblem*:

$$\max \sum_{k\in K}\Big(\sum_{p\in P^D_{pax}(k)} \sum_{e\in E(pax)} r^k_p y^{k,e}_p + \sum_{\substack{i(p)\in PI^T_{pax}(k),\\ p\in P^T(k)}} \sum_{e\in E(pax)} r^k_{i(p)} y^{k,e}_{i(p)} \Big) + \sum_{k\in K}\Big(\sum_{p\in P^D_{frt}(k)} r^k_p y^{k,f}_p + \sum_{\substack{i(p)\in PI^T_{frt}(k)\\ p\in P^T(k)}} r^k_{i(p)} y^{k,f}_{i(p)} \Big) \tag{4.13}$$

subject to:

$$\sum_{k\in K} \sum_{p\in P^D_{pax}(k)} y^{k,e}_p \delta^p_i + \sum_{k\in K,\, i(p)\in PI^T_{pax}(k)} \sum_{p\in P^T(k)} y^{k,e}_{i(p)} \delta^{i(p)}_i + \sum_{k\in K,\, o(p)\in PO^T_{pax}(k)} \sum_{p\in P^T(k)} y^{k,e}_{o(p)} \delta^{o(p)}_i \le \bar{x}^e_i d^e, \forall i \in L(pax), e\in E(pax) \tag{4.14}$$

$$\sum_{k\in K} \sum_{p\in P^D_{frt}(k)} y^{k,f}_p \delta^p_i + \sum_{k\in K,\, i(p)\in PI^T_{frt}(k)} \sum_{p\in P^T(k)} y^{k,f}_{i(p)} \delta^{i(p)}_i + \sum_{k\in K\; o(p)\in PO^T_{frt}(k)} \sum_{p\in P^T(k)} y^{k,f}_{o(p)} \delta^{o(p)}_i \le d^f, \forall i \in L(frt) \tag{4.15}$$

$$\sum_{p\in P^D_{frt}(k)} y^{k,f}_p + \sum_{p\in P^D_{pax}(k)} \sum_{e\in E(pax)} y^{k,e}_p + \sum_{\substack{i(p)\in PI^T_{frt}(k)\\ p\in P^T(k)}} y^{k,f}_{i(p)} + \sum_{\substack{i(p)\in PI^T_{pax}(k)\\ p\in P^T(k)}} \sum_{e\in E(pax)} y^{k,e}_{i(p)} \le B^k, \forall k\in K \tag{4.16}$$

$$\left\{\begin{array}{l} \sum_{e\in E(pax)} y^{k,e}_{i(p)} = \sum_{e\in E(pax)} y^{k,e}_{o(p)}, \text{if } i(p) \in PI^T_{pax}(k) \text{ and } o(p)\in PO^T_{pax}(k) \\ \sum_{e\in E(pax)} y^{k,e}_{i(p)} = y^{k,f}_{o(p)}, \text{if } i(p) \in PI^T_{pax}(k) \text{ and } o(p)\in PO^T_{frt}(k) \\ y^{k,f}_{i(p)} = \sum_{e\in E(pax)} y^{k,e}_{o(p)}, \text{if } i(p) \in PI^T_{frt}(k) \text{ and } o(p)\in PO^T_{pax}(k) \\ y^{k,f}_{i(p)} = y^{k,f}_{o(p)}, \text{if } i(p) \in PI^T_{frt}(k) \text{ and } o(p)\in PO^T_{frt}(k) \end{array}\right\}, \forall p\in P^T(k) \tag{4.17}$$

$$y^{k,e}_p \ge 0, \forall p\in P^D_{pax}(k), k\in K, e\in E(pax); y^{k,f}_p \ge 0, \forall p \in P^D_{frt}(k), k\in K;$$
$$y^{k,e}_{i(p)}, y^{k,e}_{o(p)} \ge 0, \forall i(p)\in PI^T_{pax}(k), o(p)\in PO^T_{pax}(k), p\in P^T(k), k\in K, e\in E(pax);$$
$$y^{k,f}_{i(p)}, y^{k,f}_{o(p)} \ge 0, \forall i(p)\in PI^T_{frt}(k), o(p)\in PO^T_{frt}(k), p\in P^T(k), k\in K.$$

After the values of passenger the fleet assignment variables are fixed, the cargo capacity of every passenger leg is fixed accordingly. With the knowledge of the fleet assignment, the infeasible paths are found and excluded from the model. For a feasible path p, at most one of the flow variables $y_p^{k,e}$ for all e or $y_p^{k,f}$ is non-zero, because only one fleet type is assigned to a leg. They are combined together and replaced by a single flow variable y_p^k, which is defined for every feasible path. Similarly, since all but one capacity constraints defined for a passenger leg have zero right hand sides, we aggregate them together to form a single capacity constraint. Constraints (4.17) are removed because the transferring paths need not to be split any more once the fleet type is determined and the feasibility of the path is known. The variables corresponding to subpaths *i(p)* and *o(p)* are also eliminated. Furthermore, it is no longer necessary to differentiate the direct path and transferring path.

After these simplifications the primal subproblem becomes equivalent to the individual cargo routing model of Section 3.2.4, except that the capacity constraints are defined for passenger and freighter legs separately. Note that the set of potential paths $P(k)$ for commodity k is replaced by $P_f(k)$, the feasible path set for this commodity. Instead of (4.13)-(4.17), the following simplified subproblem will be used to construct Benders cuts.

$$\max \sum_{k \in K} \sum_{p \in P_f(k)} r_p^k y_p^k \tag{4.18}$$

subject to:

$$\sum_{k \in K} \sum_{p \in P_f(k)} y_p^k \delta_i^p \leq \sum_{e \in E(pax)} (\bar{x}_i^e d^e), \forall i \in L(pax), \tag{4.19}$$

$$\sum_{k \in K} \sum_{p \in P_f(k)} y_p^k \delta_i^p \leq d^f, \forall i \in L(frt), \tag{4.20}$$

$$\sum_{p \in P_f(k)} y_p^k \leq B^k, \forall k \in K \tag{4.21}$$

$$y_p^k \geq 0, \forall p \in P_f(k), k \in K$$

Let $\pi_i^{pax}, i \in L(pax)$, $\pi_i^{frt}, i \in L(frt)$ and $\sigma^k, k \in K$ be the dual variables associated with constraints (4. 19), (4.20), and (4.21), respectively. The *dual subproblem* is written as:

$$\min \sum_{i \in L(pax)} \sum_{e \in E(pax)} (\bar{x}_i^e d^e) \pi_i^{pax} + d^f \sum_{i \in L(frt)} \pi_i^{frt} + \sum_{k \in K} B^k \sigma^k \tag{4.22}$$

subject to:

$$\sum_{i \in L(pax)} \delta_i^p \pi_i^{pax} + \sum_{i \in L(frt)} \delta_i^p \pi_i^{frt} + \sigma^k \geq r_p^k, \forall p \in P_f(k), k \in K \tag{4.23}$$

$$\pi_i^{pax}, \pi_i^{frt}, \sigma^k \geq 0, \forall i \in L(pax) \cup L(frt), k \in K$$

The primal subproblem (4.18)-(4.21) is always feasible and bounded for any given fleet assignment. We can set all variables to zeros as one feasible solution. The objective value will be at most the unconstrained revenue. Therefore, no Benders feasibility cuts are required. At each Benders iteration the dual subproblem will generate a Benders optimality cut in the form of (4.24) for one extreme point of the dual polyhedron until the optimality is reached. Note that the feasible region of the dual subproblem does not depend on the fleet assignment solution $\bar{x}_i^e, i \in L(pax), e \in E(pax)$, which will only affect the dual objective function.

$$\eta \leq \sum_{i \in L(pax)} \sum_{e \in E(pax)} (x_i^e d^e) \bar{\pi}_i^{pax} + d^f \sum_{i \in L(frt)} \bar{\pi}_i^{frt} + \sum_{k \in K} B^k \bar{\sigma}^k \tag{4.24}$$

where $(\overline{\boldsymbol{\pi}}^{\mathbf{pax}}, \overline{\boldsymbol{\pi}}^{\mathbf{frt}}, \overline{\boldsymbol{\sigma}})$ is an extreme point of the dual polyhedron Q, defined by $Q = \begin{Bmatrix} \sum_{i \in L(pax)} \delta_i^p \pi_i^{pax} + \sum_{i \in L(frt)} \delta_i^p \pi_i^{frt} + \sigma^k \geq r_p^k, \forall p \in P_f(k), k \in K \\ \pi_i^{pax}, \pi_i^{frt}, \sigma^k \geq 0, \forall i \in L(pax) \cup L(frt), k \in K \end{Bmatrix}$. Let $\mathbf{P_Q}$ be the set of extreme points of Q.

After introducing the additional free variable η, the integrated mixed integer model (3.8)-(3.15) can be reformulated as the following Benders master problem:

$$\max \eta - \sum_{i \in L(pax)} \sum_{e \in E(pax)} c_i^e x_i^e \tag{4.25}$$

subject to:

$$\eta \leq \sum_{i \in L(pax)} \sum_{e \in E(pax)} (x_i^e d^e) \overline{\pi}_i^{pax} + \sum_{i \in L(frt)} d^f \overline{\pi}_i^{frt} + \sum_{k \in K} B^k \overline{\sigma}^k, \quad ((\overline{\boldsymbol{\pi}}^{\mathbf{pax}}, \overline{\boldsymbol{\pi}}^{\mathbf{frt}}, \overline{\boldsymbol{\sigma}}) \in \mathbf{P_Q}) \tag{4.26}$$

$$\sum_{e \in E(pax)} x_i^e = 1, \forall i \in L(pax) \tag{4.27}$$

$$g_{e,s,(t_{n-1},t_n)} + \sum_{i \in I(n)} x_i^e = g_{e,s,(t_n,t_{n+1})} + \sum_{i \in O(n)} x_i^e, \forall n \in N(pax), e \in E(pax) \tag{4.28}$$

$$\sum_s g_{e,s,(t_m,t_1)} + \sum_{i \in O(e)} x_i^e \leq Num^e, \forall e \in E(pax) \tag{4.29}$$

$$x_i^e \in \{0,1\}, \forall i \in L(pax), e \in E(pax)$$
$$g_{e,s,(t_n,t_{n+1})} \geq 0, \forall e \in E(pax), s \in S, t_n, t_{n+1} \in T$$

A Benders *relaxed master problem* is defined by (4.25), (4.27)-(4.29) with only a subset of Benders optimality cuts (4.26).

4.3 Basic Algorithm

This algorithm is developed based on Benders decomposition. We iterate to solve a relaxed master problem and a subproblem. The solution of the relaxed master problem determines a feasible fleet assignment that is used to update the columns and rows in the subproblem or the cargo routing model. As we have discussed above, the feasible paths change with the fleet assignment. Because of the huge number of paths, generating them dynamically is very expensive. Instead, we only generate the potential feasible paths at the beginning of the algorithm according to the non-fleeted schedule. Then at each iteration we check the fleet type of legs in the paths to exclude from the model those infeasible ones under the current fleet assignment decision. The specific solution approach is described below.

Initialization: Generate all potential feasible paths based on the non-fleeted flight schedule including the passenger and freighter networks. Set the upper bound $UB \leftarrow \infty$, lower bound $LB \leftarrow -\infty$ and choose a relative optimality tolerance ε. Let H=0 represent the iteration number.

Step 0: Solve the initial Benders relaxed master problem, the basic FAM, to integer optimality by Branch-and-Cut. We obtain a set of fleet assignment $\bar{\mathbf{x}}^H$.

Step 1: For the current fleet assignment decision $\bar{\mathbf{x}}^H$, check the fleet type of all potential feasible paths and only add those valid to the primal subproblem. As we have discussed, the primal subproblem must have an optimal solution. Let $\bar{\mathbf{y}}^H$ be a solution and $v(\bar{\mathbf{x}}^H)$ the objective value. If the current LB is less than $(v(\overline{\mathbf{x}}^H) - \sum\sum c_i^e \overline{x}_i^{eH})$, update *LB* by this quantity and store $(\overline{\mathbf{x}}^H, \overline{\mathbf{y}}^H)$ as the incumbent solution. If $(UB - LB)/LB < \varepsilon$, terminate and the optimal solution is obtained. Otherwise determine an optimal dual subproblem solution and generate a Benders optimality cut to the relaxed master problem. Increase H by 1 and go to Step 2.

Step 2: Re-optimize the relaxed master problem to integer optimality by Branch-and-Cut. Store the optimal solution $\overline{\mathbf{x}}^H$. Set the optimal value to be the new *UB*. If $(UB - LB)/LB < \varepsilon$, terminate and the optimal solution of the original problem is obtained; otherwise go to Step 1.

This algorithm uses a relative ε optimal termination criterion. The incumbent solution $(\overline{\mathbf{x}}^H, \overline{\mathbf{y}}^H)$ is demonstrated to be ε-optimal when the relative difference of the available upper and lower bounds on the optimal value of the original problem is within ε upon termination. Prior to termination the incumbent solution is known only within *(UB-LB)/LB* of the optimal value.

In the worst cases all Benders cuts will be enumerated. Applications of Benders decomposition to the practical problems are not universally successful. Magnanti and Wong (1981) reported a case of very slow convergence of Benders decomposition when applied to network design problems. The major computational bottleneck is the huge number of integer programs to be solved. Many suggestions have been made to accelerate the Benders algorithm. For example, Magnanti and Wong (1981) proposed to choose strong cuts at each iteration, if the dual subproblem has multiple optima. Geoffrion and

Graves (1974) suggested an ε-optimal method where the relaxed master problem was not solved to optimality, but instead stopped at good feasible integer solutions. To further explore the performance of Benders decomposition and find a fast way to obtain convergence, we develop two other solution approaches based on these acceleration techniques.

4.4 Pareto-Optimal Cut Generation Approach

The concepts of *dominance* and *pareto-optimality* were introduced by Magnanti and Wong (1981) for the following general problem.

$$\begin{aligned}
&\min\ z \\
&s.t. \\
&z \geq f(u) + yg(u), \forall u \in U \\
&z \in R, y \in Y
\end{aligned}$$

The cut

$$z \geq f(u^1) + yg(u^1)$$

is stronger than or *dominates* the cut

$$z \geq f(u^2) + yg(u^2)$$

iff

$$f(u^1) + yg(u^1) \geq f(u^1) + yg(u^1)$$

holds for all $y \in Y$ with a strict inequality for at least one point $y_0 \in Y$.

A cut is called *pareto-optimal* if it is not dominated by any other cut. Since any point $u \in U$ determines a Benders cut, u^1 is said to dominate u^2 if the associated cut is stronger, and u is said pareto-optimal if the corresponding cut is pareto optimal.

If a selection of Benders cuts is possible, the judicious choice of the dual solution and the corresponding Benders cut will affect the convergence speed. Our subproblem is a multicommodity network flow problem, which is generally degenerate and its dual has alternate optimal solutions. Thus the strong cut selection method is applicable to our problem. Magnanti and Wong (1981) introduced a linear programming model in their works to generate the pareto-optimal cut. We derive this auxiliary model for our problem as:

$$\min \sum_{i \in L(pax)} \sum_{e \in E(pax)} (x_i^{e,0} d^e) \pi_i^{pax} + d^f \sum_{i \in L(frt)} \pi_i^{frt} + \sum_{k \in K} B^k \sigma^k \tag{4.30}$$

subject to:

$$\sum_{i \in L(pax)} \delta_i^p \pi_i^{pax} + \sum_{i \in L(frt)} \delta_i^p \pi_i^{frt} + \sigma^k \geq r_p^k, \forall p \in P_f(k), k \in K \tag{4.31}$$

$$\sum_{i \in L(pax)} \sum_{e \in E(pax)} (\bar{x}_i^e d^e) \pi_i^{pax} + d^f \sum_{i \in L(frt)} \pi_i^{frt} + \sum_{k \in K} B^k \sigma^k = v(\bar{\mathbf{x}}) \tag{4.32}$$

$$\pi_i^{pax}, \pi_i^{frt}, \sigma^k \geq 0, \forall i \in L(pax) \cup L(frt), k \in K$$

$\mathbf{x}^0 = \{x_i^{e,0}\}$, a relative interior point, is any point contained in the relative interior of the convex hull of the solution set $\{(x_i^e)\}$. $\bar{\mathbf{x}} = \{\bar{x}_i^e\}$ solves the current relaxed master problem and $v(\bar{\mathbf{x}})$ is the optimal objective value of the current subproblem. This formulation uses the dual variables and is in the dual form. It is quite similar to the dual subproblem (4.22)-(4.23), but with one more constraint (4.32) and different objective coefficients for $\pi_i^{pax}, i \in L(pax)$. The solution of this program defines a pareto-optimal cut.

Instead of solving this model, we are interested in solving its primal formulation, which can be easily built from the primal subproblem that we use in the basic algorithm. Let y_p^k and w be the dual corresponding to the two sets of constraints respectively. The primal formulation is:

$$\max \sum_{k \in K} \sum_{p \in P_f(k)} r_p^k y_p^k + v(\bar{x}) w \tag{4.33}$$

subject to.

$$\sum_{k \in K} \sum_{p \in P_f(k)} y_p^k \delta_i^p + w \cdot \sum_{e \in E(pax)} (\bar{x}_i^e d^e) \leq \sum_{e \in E(pax)} (x_i^{e,0} d^e), \forall i \in L(pax), \tag{4.34}$$

$$\sum_{k \in K} \sum_{p \in P_f(k)} y_p^k \delta_i^p + w \cdot d^f \leq d^f, \forall i \in L(frt), \tag{4.35}$$

$$\sum_{p \in P_f(k)} y_p^k + w \cdot B^k \leq B^k, \forall k \in K \tag{4.36}$$

$$y_p^k \geq 0, \forall p \in P_f(k), k \in K.w, free$$

The solution approach only differs from the basic algorithm in the cut generation in step 1. If $(UB - LB)/LB > \varepsilon$, instead of constructing a cut by the dual subproblem solution, the auxiliary linear model (4.33)-(4.36) is built with the current solutions of the relaxed master problem and the subproblem. Solve this model and obtain its dual solution. The pareto-optimal Benders cut is then constructed from this dual solution and is added into the relaxed master problem. In comparison with the basic approach, this procedure must solve one more linear programming problem at every iteration. However, the strong cuts will tighten the relaxed master problem or the upper bound and therefore reduce the number of iterations required to reach the optimality. The total convergence time depends on the combined effect of the reduced iteration number and the increased solution time of each iteration. According to the tradeoff, we can choose to generate pareto-optimal cuts at every iteration, or possibly, to generate cuts only periodically in the implementation of the strong cut generation method.

4.5 ε-Optimal Approach

Given any feasible fleet assignment, a Benders cut can be obtained by solving the resulting subproblem. This observation motivates a method that solves the relaxed master problem not to integer optimality, but rather stops as soon as a feasible integer solution is produced. This method alleviates the burden of optimizing an exorbitant number of integer programming problems, at the cost of the increased number of iterations required because of the weak Benders cut generated at each iteration. Geoffrion and Graves (1974) proposed such a modified Benders algorithm, where the relaxed master problem was solved to only a feasible solution with the objective value beyond the lower bound plus a tolerance ε. This means the solution of the relaxed master problem no longer defines an upper bound on the optimal value of the original problem, and the termination criteria of $UB - LB \leq \varepsilon$ should be inactivated. Instead, the algorithm terminates whenever the relaxed master has no feasible solution beyond $LB + \varepsilon$. It still converges to an ε-optimal solution within a finite number of iterations. For this reason, this technique is referred to as the ε-*optimal method* (Magnanti and Wong, 1990).

We use a relative tolerance in our optimality test, thus the termination criterion becomes that the relaxed master problem has no feasible solution beyond $LB(1+\varepsilon)$. On our problem, the ε-optimal approach constructs the following relaxed master problem at iteration *H*.

$$\max \phi(x) \tag{4.37}$$

subject to:

$$\sum_{e \in E(pax)} x_i^e = 1, \forall i \in L(pax) \tag{4.38}$$

$$g_{e,s,(t_{n-1},t_n)} + \sum_{i \in I(n)} x_i^e = g_{e,s,(t_n,t_{n+1})} + \sum_{i \in O(n)} x_i^e, \forall n \in N(pax), e \in E(pax) \tag{4.39}$$

$$\sum_s g_{e,s,(t_m,t_1)} + \sum_{i \in O(e)} x_i^e \le Num^e, \forall e \in E(pax) \tag{4.40}$$

$$\begin{aligned} LB^{(h)}(1+\varepsilon) \le & \sum_{i \in L(pax)} \sum_{e \in E(pax)} (x_i^e d^e) \overline{\pi}_i^{pax(h)} + d^f \sum_{i \in L(frt)} \overline{\pi}_i^{frt(h)} + \sum_{k \in K} B^k \overline{\sigma}^{k(h)} \\ & - \sum_{i \in L(pax)} \sum_{e \in E(pax)} c_i^e x_i^e, ((\overline{\boldsymbol{\pi}}^{\mathbf{pax}(h)}, \overline{\boldsymbol{\pi}}^{\mathbf{frt}(h)}, \overline{\boldsymbol{\sigma}}^{(h)}) \in \mathbf{P_Q}, h = 1,...,H) \end{aligned} \tag{4.41}$$

$LB^{(h)}$ is the lower bound at iteration *h*, and $(\overline{\boldsymbol{\pi}}^{\mathbf{pax}(h)}, \overline{\boldsymbol{\pi}}^{\mathbf{frt}(h)}, \overline{\boldsymbol{\sigma}}^{(h)})$ is the dual subproblem solution at this iteration. Benders Cuts (4.41) are derived by eliminating η from the following two expressions.

$$\begin{cases} LB(1+\varepsilon) \le \eta - \sum_{i \in L(pax)} \sum_{e \in E(pax)} c_i^e x_i^e \\ \eta \le \sum_{i \in L(pax)} \sum_{e \in E(pax)} (x_i^e d^e) \overline{\pi}_i^{\,pax} + d^f \sum_{i \in L(frt)} \overline{\pi}_i^{\,frt} + \sum_{k \in K} B^k \overline{\sigma}^k \end{cases}$$

Since the relaxed master problem becomes feasibility seeking only, its objective function can take any form as long as it enables the production of useful feasible solutions that can greatly improve the lower bound and therefore accelerate the convergence. According to Geoffrion and Graves (1974), the expression of the right hand side of the latest cut of (4.41) is chosen as the current objective function $\phi(x)$. The corresponding solution approach works as follows.

Initialization: Generate all potential feasible paths based on the non-fleeted flight schedule including the passenger and freighter networks. Set the lower bound $LB \leftarrow -\infty$ and choose the relative convergence tolerance ε. Let H=0 represent the iteration number.

Step 0: Solve the initial Benders relaxed master problem—the basic FAM to integer optimality by Branch-and-Cut. We obtain a set of fleet assignment $\overline{\mathbf{x}}^H$.

Step 1: For the current fleet assignment decision $\overline{\mathbf{x}}^H$, check the fleet type of all potential feasible paths and only add those valid to the Benders primal subproblem. Let $\overline{\mathbf{y}}^H$ be the solution of the subproblem and $v(\overline{\mathbf{x}}^H)$ the objective value. If the current LB is less than $(v(\overline{\mathbf{x}}^H) - \sum\sum c_i^e \overline{x}_i^{eH})$, update LB by this quantity and store $(\overline{\mathbf{x}}^H, \overline{\mathbf{y}}^H)$ as the incumbent solution. Determine an optimal dual subproblem solution and generate one Benders cut to the relaxed master problem. Increase H by 1 and go to Step 2.

Step 2: Update the objective function of the relaxed master problem as the latest cut's right hand side function. Solve it to the first integer feasible solution found, denoted by $\overline{\mathbf{x}}^H$. If there is no feasible solution, terminate and the optimal solution of original problem is obtained. Otherwise go to Step 1.

The algorithm will terminate at step 2 whenever the relaxed master problem is infeasible, with the ε-optimal solution $(\overline{\mathbf{x}}^H, \overline{\mathbf{y}}^H)$.

5 Computational Results

We have presented a Benders decomposition approach and its two variants to solve the integrated problem. To measure and compare their performances, computational experiments were performed on a set of test instances based on available data about the airline. We first describe in Section 5.1 the test instances used, and then in Section 5.2 report results of different solution approaches, including an improved method to enhance the ε-optimal approach. A comprehensive comparison between performances of these approaches is presented in Section 5.3.

5.1 Description of Data Sets

The weekly flight schedule of the airline contains 1,404 passenger legs and 201 freighter legs, which serve 74 stations around the world. Six passenger fleets and one freighter fleet are used to cover all legs. Based on this schedule we generated five test instances, each of which contains subsets of passenger legs and freighter legs. The main difficulty is to make sure the subnetwork is balanced, that is, it has the same number of arrivals and departures at each station, and uses the aircraft within the available number. Given a current fleet assignment of the schedule, this is accomplished by selecting the legs currently assigned to the same subset of fleets. It is straightforward to construct the passenger subnetworks through this approach. The balanced freighter subnetwork, however, can not be built in this way because only one freighter fleet is employed to fly all freighter legs. To construct the subset of freighter legs, we start with the full freighter network. Then we drop the balanced islands that have the same number of arrivals and departures at some stations and their connected stations. This approach maintains the balance of the resulting freighter subnetwork.

The commodities and their demands are generated from the historical data of the airline. Based on the daily shipment between any pair of stations, we generate commodities with different shipment day, shipment time and time windows. The demand of the commodity available for shipment at the end of a business day is assumed to be higher than that of commodities at the other two time slots. Once the commodities are generated and the subnetworks are constructed, the potential feasible paths for all commodities over each subnetwork (both passenger and freighter subnetworks) are generated based on Section 3.3.2. All costs are estimated from the airline's annual financial report.

Table 5.1 describes our test data instances. The first two instances are quite small and contain only passenger legs. Instances D3 to D5 contain both passenger and freighter legs, and D5 covers the full network the airline operates. We assume all passenger fleets are available for different test instances. The commodities are generated based on the unconstrained market demand, which is independent of the network. Thus the number of commodities is the same for all 5 data instances. However, the number of potential feasible paths for all commodities varies greatly with the size of the subnetwork. For the small network, lots of commodities have no potential feasible paths, which mean they can not be shipped through this subnetwork. For the full network, however, every commodity has 2.7 potential feasible paths on average. This follows from the fact that the connection opportunities reduce quickly with the reduction of legs.

Table 5.1 The Characteristics Of Data Instances

Instance	Total Number of Passenger Legs $\lvert L(pax)\rvert$	Total Number of Freighter Legs $\lvert L(frt)\rvert$	Total Number of Passenger Fleets $\lvert E(pax)\rvert$	Total Number of Freighter types	Total Number of Commodities $\lvert K\rvert$	Total Number of Potential feasible paths $\sum_k \lvert p(k)\rvert$
D1	62	0	6	0	63,798	573
D2	102	0	6	0	63,798	2,051
D3	520	201	6	1	63,798	28,221
D4	884	151	6	1	63,798	73,880
D5(Full)	1,404	201	6	1	63,798	173,285

5.2 Computational Results

All solution approaches were coded in C++. CPLEX8.1 and Concert Technology 1.3 were employed to model and solve the relaxed master problems and the subproblems. The relaxed master problem was solved by the mixed integer optimizer that uses the Branch-and-Cut method. The cuts generated by CPLEX include clique cuts, cover cuts, disjunctive cuts, Gomory fractional cuts, etc. The dual simplex algorithm with steepest edge pricing was employed to solve the LP relaxation at each node in the Branch-and-Bound tree. A priority order was generated according to the increasing cost per coefficient count and issued to every variable to control the branching direction, when the variables have fractional values. For a column or a variable, the more the ratio of the objective coefficient over the number of nonzero entries, the lower priority value the variable will get. At each node, variables with a higher priority are branched before variables with a lower priority. The subproblem was solved by the primal simplex optimizer. All experiments were carried out on the computer with 866 MHz CPU and 256 MB of RAM. The relative convergence tolerance ε is set to 0.1%. To speed up the solving of the relaxed master problem, some CPLEX parameters that have much influence on the Branch & Cut procedure are set to appropriate values, as shown in Table 5.2.

Table 5.2 CPLEX Parameters

Parameter	Description	Set Value	Meaning
BtTol	Backtracking Tolerance	0.1	Force Branch & Cut not to dive deep into the tree.
EpGap	Relative Mip Gap Tolerance.	0.002	CPLEX stops as soon as a feasible integer solution proved to be within 0.2% of optimal.
HeurFreq	MIP Heuristic Frequency	20	Apply the heuristic to find integer solutions at every 20 nodes during branch & Cut procedure.
MIPEmphasis	MIP Emphasis Indicator	0	Emphasize balanced optimality and feasibility.
MIPOrdType	MIP Priority Order Generation	3	Use increasing cost per coefficient count.
RelObjDif	Relative Objective Difference	0.0001	Speed up the proof of optimality. CPLEX skips any potential solution with its objective value within 0.01% of the best integer solution so far.

5.2.1 Basic Algorithm

Table 5.3 reports the computational results for all test instances solved by the basic algorithm. During the experiments we found the solution time was greatly effected by the unit cargo selling price. In order to explore the underlying reason we vary this coefficient, which is reduced by 50% in data sets with one asterisk, and is increased by 50% in data sets with two asterisks.

Table 5.3 Computational Results Of The Basic Algorithm

Instances	Convergence CPU time (s)	No. Benders Iteration	Time (s) Per Iteration	Benders final Relative gap
D1	3.1	11	0.3	0.02%
D1*	0.4	1	0.4	0.00%
D1**	9.8	20	0.5	0.09%
D2	1.3	3	0.4	0.09%
D2*	0.8	2	0.4	0.00%
D2**	1.3	3	0.4	0.09%
D3	14.7	4	3.7	0.08%
D3*	7.5	2	3.7	0.02%
D3**	19.2	5	3.8	0.07%
D4	103.5	13	8.0	0.08%
D4*	45.7	6	7.6	0.08%
D4**	158.0	17	9.3	0.09%
D5	116.7	4	30.0	0.09%
D5*	107.0	3	35.7	0.09%
D5**	177.7	9	19.8	0.09%

*: Unit cargo selling price is decreased by 50%
**: Unit cargo selling price is increased by 50%

Table 5.3 shows that the number of iterations before reaching optimality is quite small and the convergence is very fast for every test instance. D2 spent only 1.3 seconds and 3 iterations to obtain the optimal solution. Even for the full instance D5, the optimality was reached within 116.7 seconds by 4 iterations. The basic algorithm is therefore proven to be efficient to our problem. One comparable data set to D5 in Barnhart et al. (2002) took 3,400 seconds (about 30 times longer than the time we spent on D5) to find the best solution, even though they used the solution enhanced key-path formulation. Our fast solution may be attributed to the single hub character of the flight network and the fast solution speed of CPLEX8.1.

Another finding is that the solution time increases/reduces with the increase/reduction of the unit cargo selling price. This phenomenon is especially obvious for the large instances D4 and D5. For D4, the number of iterations increased from 13 to 17 and the corresponding solution time increased from 105.3s to 158.0s when the price was 50% higher. Similar for D5, the increased price led to 60 seconds more to reach the optimality. On the contrary, when the price was 50% off, fewer iterations and shorter times were required for the two instances to converge. They spent only 45.7 seconds with 6 iterations and 107.0 seconds with 3 iterations, respectively.

Although the application of the basic algorithm was successful to our problem, the other two variants were still implemented in order to, if possible, further accelerate the convergence and obtain some insights on their applications to realistic problems.

5.2.2 Pareto-Optimal Cut Generation Approach

We first implemented the pareto-optimal cut generation method. A relative interior point in the convex hull of the solution set $\{(x_i^e)\}$ was found as the convex combination of six feasible integer fleet assignment solutions and one feasible fractional solution. As the total number of the available aircraft is more than that required to fly the schedule, we reduced the size of one passenger fleet by one and solved the resulting FAM, which provided a feasible fleet assignment solution. By repeating the same procedure for the other five passenger fleets we obtained the other five feasible fleet assignment solutions. The convex combination of these six feasible integer solutions will make the strict less hold for the constraints (3.3). The feasible fractional solution was obtained by assigning every variable x_i^e a value as the number of the aircraft available in the fleet e over the total number of the aircraft available in all the fleet types. Obviously this fractional solution satisfies the cover constraint (3.1). The balance constraints (3.2) are easily satisfied because for every fleet *e*, the variable x_i^e assumes the same value for every leg *i*. As only a small number of legs are crossed by the count time, the constraints (3.3) are always satisfied for this fractional solution. This fractional solution does not lie on any facet of the convex hull where the convex combination of the six integer solutions lies. As a result, the convex combination of these seven solutions does not lie on any facet and is a relative interior point in the convex hull.

The primal simplex algorithm was employed to solve the auxiliary linear programming in the primal form. Initially we found this auxiliary model was unbounded or its dual was infeasible. This was attributed to the numerical rounding error of the computer, which caused the constraint (4.32) violated. Instead of strict equality, we permitted the left hand side of (4.32) very small fluctuation within $\left[v(\bar{\mathbf{x}}), v(\bar{\mathbf{x}}) + \Delta \right]$ to accommodate this error, where Δ is a small enough positive value. The modified auxiliary model in the dual form is:

$$\min \sum_{i \in L(pax)} \sum_{e \in E(pax)} (x_i^{e,0} d^e) \pi_i^{pax} + d^f \sum_{i \in L(frt)} \pi_i^{frt} + \sum_{k \in K} B^k \sigma^k \tag{5.1}$$

subject to:

$$\sum_{i \in L(pax)} \delta_i^p \pi_i^{pax} + \sum_{i \in L(frt)} \delta_i^p \pi_i^{frt} + \sigma^k \geq r_p^k, \forall p \in P_f(k), k \in K \tag{5.2}$$

$$\sum_{i \in L(pax)} \sum_{e \in E(pax)} (\bar{x}_i^e d^e) \pi_i^{pax} + d^f \sum_{i \in L(frt)} \pi_i^{frt} + \sum_{k \in K} B^k \sigma^k \leq v(\bar{X}) + \Delta \tag{5.3}$$

$$\pi_i^{pax}, \pi_i^{frt}, \sigma^k \geq 0, \forall i \in L(pax) \cup L(frt), k \in K$$

Let w be the dual variable corresponding to constraint (5.3). The primal model becomes:

$$\max \sum_{k \in K} \sum_{p \in P_f(k)} r_p^k y_p^k + (v(\bar{X}) + \Delta) w \tag{5.4}$$

subject to:

$$\sum_{k \in K} \sum_{p \in P_f(k)} y_p^k \delta_i^p + w \sum_{e \in E(pax)} (\bar{x}_i^e d^e) \le \sum_{e \in E(pax)} (x_i^{e,0} d^e), \forall i \in L(pax) \tag{5.5}$$

$$\sum_{k \in K} \sum_{p \in P_f(k)} y_p^k \delta_i^p + w \cdot d^f \le d^f, \forall i \in L(frt) \tag{5.6}$$

$$\sum_{p \in P_f(k)} y_p^k + w \cdot B^k \le B^k, \forall k \in K \tag{5.7}$$

$$y_p^k \ge 0, \forall p \in P_f(k), k \in K.$$
$$w \le 0$$

The computational results by this strong cut selection approach are reported in Table 5.4.

Table 5.4 Computational Results Of
Pareto Optimal Cut Generation Approach

Instances	Convergence CPU time (s)	No. Benders Iterations	Time (s) per Iteration	Benders Final Relative Gap
D1	4.2	11	0.4	0.02%
D1*	0.6	1	0.6	0.00%
D1**	14.3	20	0.7	0.09%
D2	1.9	3	0.6	0.09%
D2*	1.6	2	0.8	0.00%
D2**	2.4	3	0.8	0.09%
D3	36.8	4	9.2	0.09%
D3*	20.6	2	10.3	0.05%
D3**	50.3	5	10.2	0.05%
D4	870.9	11	79.2	0.06%
D4*	534.2	6	89.0	0.08%
D4**	1394.3	18	77.5	0.01%
D5	2254.2	5	450.8	0.07%
D5*	1105.5	2	552.8	0.07%
D5**	4367.4	9	485.3	0.09%

*: Unit cargo selling price is decreased by 50%
**: Unit cargo selling price is increased by 50%

By comparing the results with those in Table 5.3, we found that there was no improvement with the selection of strong cuts. On the contrary, the convergence became even slower for all test instances. The number of iterations remained the same for D1 to D3, but their solution times were increased. The only exception was D4, which spent 2 less iterations to generate a better solution. D5 was the worst case where the number of iterations increased by 1. This extra iteration was used to improve the final solution with a relative gap at 0.07%, compared with 0.09% of the solution by the basic algorithm. No matter what changes in the iteration number, the average time per iteration increased greatly, especially for D4 and D5. The main reason is that at each iteration, one more linear program must be solved, and the solution time increases very quickly with the problem size. For example, the full instance D5 took 250 seconds on average to solve an auxiliary model at each iteration, more than half of the total time of the iteration.

Similar to the results in Table 5.3, the solution time increases/reduces with the increase/reduction of the unit cargo selling price. However, every instance spent longer time to converge than their counterparts did by the basic algorithm, even though the number of iterations required was almost the same.

To explore the reason why the performance of the pareto-optimal cut generation method is so poor in our problem, we compared two cuts from instance D4 for a series of feasible fleet assignments solutions. Cut1 is directly generated from the dual solution of a subproblem, while cut2 from the dual solution of the corresponding auxiliary model. For each feasible fleet assignment solution $\bar{\mathbf{x}}$, the values of the two cuts' right hand side, $\sum_{i\in L(pax)}\sum_{e\in E(pax)}(\bar{x}_i^e d^e)\bar{\pi}_i^{pax} + d^f \sum_{i\in L(frt)}\bar{\pi}_i^{frt} + \sum_{k\in K} B^k \bar{\sigma}^k$, were computed and listed in Table 5.5.

Table 5.5 Comparison Of Two Cuts' Right Hand Values

Index of Feasible Fleet Assignment Solution	RHS Value of Cut1	RHS Value of Cut2	Absolute Difference	Relative Difference
1	20987129.26	20987121.42	-7.839299999	-0.00004%
2	20421106.28	20421105.7	-0.574999999	0.00000%
3	20607237.96	20607234.24	-3.7172	-0.00002%
4	20552858.41	20552857.84	-0.574700002	0.00000%
5	20540869.37	20540869.13	-0.239699997	0.00000%
6	20584393.32	20584391.31	-2.017000001	-0.00001%
7	20594521.32	20594521.08	-0.2355	0.00000%
8	20655740.98	20655739.48	-1.506299999	-0.00001%
9	20609321.88	20609318.92	-2.953700002	-0.00001%
10	20669920.26	20669917.31	-2.9549	-0.00001%
11	20514162.91	20514160.63	-2.279200003	-0.00001%

Table 5.5 shows that although the RHS of cut2 is always less than that of cut1, the difference is too small to justify the strong cut2. The relative differences are almost zero for all cases so that there is not much difference between these two cuts. We have checked the dual solutions corresponding to these two cuts and found that they were definitely different. This fact excluded the possibility that the two cuts were constructed from the same dual solution and the small difference between them was caused only by the round off errors. The effect of the strong cut generation, therefore, is not significant in our problem. We checked the degeneracy degree (the percent of basic variables at or near zero in the basic solution) of the subproblem, which is at most 15%. This low degree of degeneracy may be a result of the fractional values of the right hand sides of the demand constraints and the big variation among them. In this case, the excess number (over the number of variables) of constraints or hyperplanes passing through an extreme point is

small, and thus the degeneracy is not severe. As a result, the alternate dual subproblem solutions only differ a little from each other, causing the cuts defined by them to be almost the same. The extra time of selecting such a “strong” cut is not offset by its quality.

5.2.3 ε-Optimal Approach

Set the parameter “MIPEmphasis” in Table 5.2 to “1”, emphasizing feasibility. The results of this variant are shown in Table 5.6. Since the upper bound is not available in this method, the relative gap between the upper and lower bounds can not be obtained.

Table 5.6 Computational Results
Of The ε-Optimal Approach

Instances	Convergence CPU time (s)	No. Benders Iteration	Time (s) per Iteration
D1	5.4	18	0.3
D1*	0.4	1	0.4
D1**	24.4	39	0.6
D2	1.4	3	0.5
D2*	0.9	2	0.5
D2**	1.4	3	0.5
D3	14.8	5	3.0
D3*	7.4	2	3.7
D3**	16.4	5	3.3
D4	Does not converge in 24 hours		
D4*	6600.0	37	178.4
D4**	Does not converge in 24 hours		
D5	565.0	14	40.4
D5*	121.6	4	30.4
D5**	3423.8	17	201.4

*: Unit cargo selling price is decreased by 50%
**: Unit cargo selling price is increased by 50%

The comparatively small instances D1 to D3 converged quite fast. Although more iterations, with respect to the results in Table 5.3 and Table 5.4, were spent to reach the optimality, they were compensated by the reduced solution time of each iteration. For the full instance D5, both the number of iterations and the time per iteration increased, which resulted in the total solution time being several times more than that by the basic algorithm. Even worse, the instance D4 can not converge in 24 hours. However, it found an optimal solution in 6600s with 37 iterations when the unit cargo selling price was reduced by 50%.

After observing the computing process of D4 and D5, we found that most time was spent in the last several iterations, especially the last one, which took extremely long time to prove the infeasibility of the relaxed master problem. To find ways overcoming this difficulty, we tried to relax the relative convergence tolerance ε gradually from 0.1% to 0.5%. The corresponding results for D4 are described in Table 5.7.

Table 5.7 Results Of D4 With
The Different Relative Convergence Tolerance

ε (%)	0.5	0.4	0.3	0.2	0.19	0.18	0.15
Number of iteration	4	4	7	15	14	25	30
Convergence time (s)	35	35	66	180	174	17,144	6,534
Time of the last iteration (s)	1	1	2	10	6	12,700	3,900

It is very clear that the relative convergence tolerance has a significant influence on the solution time. When ε is greater than 0.2% the ε-optimal approach works well, but it suddenly deteriorates once ε becomes less than 0.19%. For the small tolerance, the algorithm spent more than 2/3 of the total solution time to prove the last relaxed master problem had no feasible solution. On the contrary, very short time was required to accomplish it when ε was large. This phenomenon is not obvious for small problems like D1 to D3 because they are relatively easy to solve.

5.2.4 Proposed Hybrid Approach

Although relaxing ε accelerates the convergence, the solution quality is compromised. In order to generate good solutions quickly, we suggest a hybrid approach. First the ε-optimal approach is employed to find a good feasible solution, where ε is set to a larger value. After that, we decrease ε and turn to the basic algorithm to generate the solution closer to the optimality. At each iteration in phase1 two Benders cuts are constructed from the same dual subproblem solution. One cut is in the form (4.41) and is added to the relaxed master problem of the ε-optimal approach, while the other cut is in the form (4.24) and is not used in phase1. Instead, it is retained and utilized by the basic algorithm in phase2. This approach takes advantage of the quick solution by the ε-optimal approach in the early iterations, and eliminates the burden of proving infeasibility in the last iterations.

We implemented this hybrid solution approach and set ε to be 0.5% and 0.1% in the two phases, respectively. The results are described in Table 5.8, where the values inside parentheses are the results of phase 1.

Table 5.8 Computational Results Of The Hybrid Approach.

Instances	Convergence CPU time (s)	No. Benders Iterations	Time (s) per Iteration	Benders Final Relative Gap
D1	(2.0) 3.7	(7) 11	0.3	0.03%
D1*	(0.5) 0.5	(1) 1	0.5	0.00%
D1**	(3.5) 13.7	(9) 19	0.7	0.09%
D2	(0.9) 1.4	(2) 3	0.5	0.09%
D2*	(1.1) 1.1	(2) 2	0.6	0.00%
D2**	(1.0) 1.6	(2) 3	0.5	0.09%
D3	(7.3) 15.6	(2) 5	3.0	0.08%
D3*	(9.1) 9.1	(2) 2	4.6	0.02%
D3**	(7.3) 15.3	(2) 4	3.8	0.06%
D4	(35.3) 76.1	(4) 9	8.5	0.03%
D4*	(20.4) 50.7	(2) 6	8.5	0.08%
D4**	(57.5) 302.9	(6) 28	10.8	0.08%
D5	(140.2)178.6	(4) 5	35.7	0.09%
D5*	(87.0)) 123.7	(2) 3	41.2	0.09%
D5**	(146.3) 416.8	(4) 10	41.7	0.01%

*: Unit cargo selling price is decreased by 50%

**: Unit cargo selling price is increased by 50%

It is shown that the convergence of D4 and D5 was accelerated greatly. D5 took only 5 iterations and 178.6 seconds to reach the optimality, compared with 14 iterations and 565.0 seconds in Table 5.7. For D4 that can not converge by the ε-optimal approach, the 0.03%-optimal solution was generated in 76.1 seconds, which was even shorter than that (103s) by the basic algorithm. Similarly, it took D4** 28 iterations and 302.9 seconds to reach the optimality. The improvement for D1 to D3 is negligible. The hybrid approach, therefore, works well especially for the large instance that has a difficult mixed integer relaxed master problem. One critical step in this hybrid approach is to choose an appropriate ε or the turning point between the two phases. If ε is too big, the approach turns to phase2 very early and is more like the basic algorithm, and vice versa. Hence, by judiciously setting ε we can take full advantage of the strengths of both the ε-optimal approach and the basic algorithm.

5.3 Comparison of the Four Solution Approaches

All above reported computational results by the four solution approaches are summarized in the three tables below. Table 5.9, Table 5.10, and Table 5.11 list the results with the original, 50% off and 50% higher unit cargo selling price, respectively.

Among all four solution approaches, the basic algorithm spent the least time and smallest number of iterations for all test instances to converge, no matter what unit cargo selling price is designated. Thus it is the best one to solve the integrated fleet assignment and cargo routing problem. What comes next is the hybrid approach, which reaches optimality by a little longer time than and almost the same number of iterations as the basic algorithm. For the instances D4 and D3** it is even better than the basic algorithm. Therefore the hybrid approach has the potential to become an efficient method to solve the integrated problem. Spending a large number of iterations and quite long time, the ε-optimal approach performs poorly in our problem. For the pareto-optimal cut generation approach, lots of time is used per iteration to solve the auxiliary model, which results a very long convergence time even though only several iterations are required.

Results also demonstrate that the unit cargo selling price has significant influence on the solution time for all instances, especially for large instances D4 and D5. They spent much more time and iterations to converge when the price is increased. In this case, the lower bound improves very slowly from an early iteration till reaching optimality. This may result from that the increased passenger revenue cannot compensate the decrease of cargo revenue, because small cargo capacity reduction may cause large loss of cargo revenue.

As a result, the summation of passenger and cargo revenue, namely the lower bound, can hardly improve. Reversely, the reduced unit cargo selling price leads to much shorter solution time and less iteration.

Table 5.9 Computational Results With The Original Unit Cargo Selling Price

Approach	Instance	Convergence CPU time (s)	No. Benders Iterations	Time (s) per Iteration	Benders Final Relative Gap
Basic Algorithm	D1	3.1	11	0.3	0.02%
	D2	1.3	3	0.4	0.09%
	D3	14.7	4	3.7	0.08%
	D4	103.5	13	8.0	0.08%
	D5	116.7	4	29.2	0.09%
Pareto-Optimal Cut Generation Approach	D1	4.2	11	0.4	0.02%
	D2	1.9	3	0.6	0.09%
	D3	36.8	4	9.2	0.09%
	D4	870.9	11	79.2	0.06%
	D5	2254.2	5	450.8	0.07%
ε-optimal Approach	D1	5.4	18	0.3	N/A
	D2	1.4	3	0.5	N/A
	D3	14.8	5	3.0	N/A
	D4	Does not converge in 24 hours			
	D5	565.0	14	40.4	N/A
Hybrid Approach	D1	(2.0) 3.7	(7) 11	0.3	0.03%
	D2	(0.9) 1.4	(2) 3	0.5	0.09%
	D3	(7.3) 15.6	(2) 5	3.1	0.08%
	D4	(35.3) 76.1	(4) 9	8.5	0.03%
	D5	(140.2) 178.6	(4) 5	35.7	0.09%

Table 5.10 Computational Results With The 50% Off Unit Cargo Selling Price

Approach	Instance	Convergence CPU time (s)	No. Benders Iterations	Time (s) per Iteration	Benders Final Relative Gap
Basic Algorithm	D1*	0.4	1	0.4	0.00%
	D2*	1.8	2	0.9	0.00%
	D3*	7.5	2	3.8	0.02%
	D4*	45.7	6	7.6	0.08%
	D5*	107.0	3	35.9	0.09%
Pareto-Optimal Cut Generation Approach	D1*	0.6	1	0.6	0.00%
	D2*	1.6	2	0.8	0.00%
	D3*	20.6	2	10.3	0.05%
	D4*	534.2	6	89.0	0.08%
	D5*	1105.5	2	552.8	0.07%
ε-optimal Approach	D1*	0.4	1	0.4	N/A
	D2*	0.9	2	0.5	N/A
	D3*	7.4	2	3.7	N/A
	D4*	6600.0	37	178.4	N/A
	D5*	121.6	4	30.4	N/A
Hybrid Approach	D1*	(0.5) 0.5	(1) 1	0.5	0.00%
	D2*	(1.1) 1.1	(2) 2	0.6	0.00%
	D3*	(9.1) 9.1	(2) 2	4.6	0.02%
	D4*	(20.4) 50.7	(2) 3	16.9	0.08%
	D5*	(87.0) 123.7	(4) 5	24.7	0.09%

Table 5.11 Computational Results With The 50% Higher Unit Cargo Selling Price

Approach	Instance	Convergence CPU time (s)	No. Benders Iterations	Time (s) per Iteration	Benders Final Relative Gap
Basic Algorithm	D1**	9.8	20	0.5	0.09%
	D2**	1.3	3	0.4	0.09%
	D3**	19.2	5	3.8	0.07%
	D4**	158.0	17	9.3	0.09%
	D5**	177.7	9	19.7	0.09%
Pareto-Optimal Cut Generation Approach	D1**	14.3	20	0.7	0.09%
	D2**	2.4	3	0.8	0.09%
	D3**	50.3	5	10.1	0.05%
	D4**	1394.3	18	77.5	0.01%
	D5**	4367.4	9	485.3	0.09%
ε-optimal Approach	D1**	24.4	39	0.8	N/A
	D2**	1.4	3	0.4	N/A
	D3**	16.4	5	3.3	N/A
	D4**	Does not converge in 24 hours			
	D5**	3423.8	17	201.4	N/A
Hybrid Approach	D1**	(3.5) 13.7	(9) 19	0.7	0.09%
	D2**	(1.0) 1.6	(2) 3	0.5	0.09%
	D3**	(7.3) 15.3	(2) 4	3.8	0.06%
	D4**	(57.5) 302.9	(6) 28	10.8	0.08%
	D5**	(146.3) 416.8	(4) 10	41.7	0.01%

6 Conclusions and Future Research

This chapter concludes this thesis in Section 6.1 and proposes directions for future research in Section 6.2.

6.1 Conclusions

Fleet assignment is the second airline schedule planning step that is made to maximize the profitability by optimally allocating fleet types to the legs. Traditionally this step ignores the cargo flow and may not fully utilize the resource of a combination carrier. The revenue contributed by cargo keeps increasing for the last decade, and hence the cargo routing should be properly modeled so as to maximize the revenue. The route of cargo is determined to a large extent by the cargo capacity of every leg, which depends on the fleet assignment decision. As a result, the fleet assignment has great influence on the cargo revenue. Incorporating the cargo routing into the fleet assignment can better balance the resource of a combination carrier and the forecasted cargo demand. Different from the passenger, cargo has no strong preference on the specific itinerary as long as its commitment is satisfied. There is also no available industry data to calculate the spill cost and the recapture rate for the cargo flow. Moreover, cargo is allowed to transfer between different aircraft only at the hub, while this requirement is not applicable to the passenger. The cargo flow is thus modeled in a way different from the passenger flow model.

Given this motivation, we proposed an integrated approach that simultaneously determines the assignment of fleet to legs and the cargo routing over the flight network. An integrated formulation combing the fleet assignment model and the cargo routing model was presented. To eliminate the complexity brought by the time window and the side constraints, a two phase technique was applied to model the cargo routing problem. The resulting cargo routing model is a path oriented MCNF, in which each column is a feasible path. Since the fleet type of every leg is determined together with the routing of cargo, the feasible path can not be generated in advance. To accommodate the uncertainty of the feasible path, we disaggregated the capacity constraints and the variables in the CRM and generated all the potential feasible paths from a non-fleeted schedule to replace feasible paths. The integrated formulation obtained is a large scale mixed integer program that contains a huge number of variables and constraints.

A Benders decomposition based algorithm was proposed to solve the integrated problem. This algorithm decomposes the integrated formulation into a relaxed master problem of the fleet assignment and a subproblem of the cargo routing. These two problems are solved iteratively until the difference between their solutions is within a designated tolerance. Since at each iteration the cargo routing model is set up and solved after the fleet assignment model, the feasible paths can be generated with the knowledge of the fleet type of every leg. As a result, the subproblem reduces to the individual CRM, whose size is much smaller than that in the integrated formulation. Other than the basic algorithm, two variants, the pareto-optimal cut generation approach and the ε-optimal approach were applied to solve the integrated problem. The pareto-optimal cut generation method selects strong cuts at each Benders iteration, while the ε-optimal approach solves the

Benders relaxed master problem only to a feasible integer solution rather than an integer optimum.

A series of computational experiments were carried out for several data sets to test and compare performances of different solution approaches. Results show that the basic algorithm worked very well and outperformed others in our problem. It took only several minutes to generate optimal solutions, which provided an improved estimate of total profit in comparison with the isolated fleet assignment. The performances of the other two variants turned out unsatisfactory. The pareto-optimal cut generation approach spent very long time to converge, even though the number of iterations required was quite small. Every iteration of it took a large amount of time to solve the auxiliary model to select a “strong” cut, which was almost the same as the cut generated directly by the subproblem. Possibly this follows from the low degeneracy degree of the primal subproblem. The main drawback of the ε-optimal approach was the infeasibility proof of the last relaxed master problem. Especially for the large instances, this part of time accounted for 2/3 of the total solution time. To overcome it a hybrid approach was suggested, which first employs the ε-optimal approach to obtain a good feasible solution, and then turns to the basic algorithm for a solution closer to the real optimum. It is shown that this hybrid approach converged very fast with few iterations. Although it was faster than the basic algorithm only in several cases, the hybrid approach has the potential to generate better results if an appropriate turning point is chosen.

6.2 Future Research

It is worth to restate that the passenger revenue is only estimated linearly in our integrated formulation. For a combination carrier passenger is still its main source of profit, thus the passenger flow problem should be properly modeled. An enhanced integrated model could also incorporate the passenger mix problem, which finds the passenger flow of maximal revenue over a given fleeted flight schedule. The resulting model will simultaneously allocate fleet types to legs and determine the flow of cargo and passengers over the network. This approach is able to balance the resource of an airline (the available cargo capacities and passenger seats) and the demands of both cargo and passengers at all markets. An improved estimate of total passenger and cargo profit is therefore expected to obtain. This extended integrated formulation can still be solved by Benders decomposition. The master problem solves the fleet assignment, and the subproblem solves the cargo routing and the passenger mix. Since the passengers' luggage will compete with cargo for the space, the models of passenger mix and cargo routing are coupled together by the cargo capacity constraints. This leads to a block-diagonal structured subproblem that can be solved by Dantzig-Wolfe decomposition. In this case, the main question to be answered is how to construct the Benders cut from the subproblem solution.

Also recall that our integrated model is just an approximation of the actual three stage problem. Another model enhancement is to incorporate the physical aircraft routing. A more comprehensive integrated model could combine all the four problems together, fleet assignment, aircraft routing, cargo routing and passenger mix. How to formulate and solve such an extremely large problem raises big challenges for future research.

The forecasted cargo demand is static in our problem. This simplification may cause the solution less convincing because the demand actually fluctuates randomly all the time. Therefore, another interesting research direction is to capture the demand uncertainty.

Also interesting is to choose a good turning point of the hybrid approach. The solution time by the ε-optimal approach changes greatly with the designated optimality tolerance. An appropriate tolerance value or a turning point could instruct the solution process to switch the basic algorithm at appropriate time, and therefore accelerate the convergence and enhance the hybrid approach.

References

Abara, J., 1989. Applying Integer Linear Programming to the Fleet Assignment Problem. *Interfaces*, 19(4): 20-28.

Ahuja, R.K., Magnanti, T.L. and Orlin, J.B., 1993. Network Flows: Theory, Algorithms and Applications. Prentice Hall, Englewood Cliffs, NJ, 649-694.

Anbil, R., Tanga, R. and Johnson, E.L., 1992. A Global Approach to Crew-Pairing Optimization. *IBM Systems Journal*, 31(1): 71-78.

Antes, J., Campen L., Derigs, U., Titze, G. and Wolle, D., 1998. SYNOPSE: A Model-Based Decision Support System for the Evaluation of Flight Schedules for Cargo Airlines. *Decision Support Systems*, 22: 307-323.

Bard, J.F. and Cunningham, L.G., 1987. Improving Through-Flight Schedules. *IIE Transactions*, 19(3): 242-251.

Barnhart, C., Boland N. L., Clarke L. W., Johnson E. L., Nemhauser G. L., Shenoi R. G., 1998a. Flight String Models for Aircraft Fleeting and Routing. *Transportation Science*, 32(3): 208-220.

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

Barnhart, C., Krishnan, N., Kim, D. and Ware, K., 2002. Network Design for Express Shipment Delivery. *Computational Optimization and Applications*, 21: 239-262.

Barnhart, C., Lu F. and Shenoi R., 1998b. “Integrated Airline Schedule Planning,” in *Operations Research in the Airline Industry*, G. Yu (Editor), 384-403, Kluwer, Boston.

Barnhart, C. and Schneur, R.R., 1996. Air Network Design for Express Shipment Service. *Operations Research*, 44(6): 852-863.

Barnhart, C. and Shenoi R. G., 1998c.“An Approximate Model and Solution Approach for the Long-Haul Crew Pairing Problem,” *Transportation Science*, 32(3), 221-231.

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problem. *Numerical Mathematics*, 4: 238-252.

Berge, M.E. and Hopperstad, C.A., 1993. Demand Driven Dispatch: A Method for Dynamic Aircraft Capacity Assignment, Models and Algorithms. *Operations Research*, 41(1): 153-168.

Clarke, L.W., Hane, C.A., Johnson, E.L. and Nemhauser, G.L., 1996. Maintenance and Crew Considerations in Fleet Assignment. *Transportation Science*, 30(3): 249-260.

Cohn, A.M. and Barnhart, C., 2003. Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research*, 51(3): 387-396.

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

Desaulniers, G., Desrosiers, J., Dumas, Y., Solomon, M.M. and Soumis, F., 1997. Daily Aircraft Routing and Scheduling. *Management Science*, 43(6): 841-855.

Geoffrion, A.M. and Graves, G.W., 1974. Multicommodity Distribution System Design by Benders Decomposition. *Management Science*, 20(5): 822-845.

Gopalan, R. and Talluri, K.T., 1998a. Mathematical Models in Airline Schedule Planning: A Survey. *Annals of Operations Research*, 96: 155-185.

Gopalan, R. and Talluri, K.T., 1998b. The Aircraft Maintenance Routing Problem. *Operations Research*, 46(2): 260-271.

Grainic, T.G., 2000. Service Network Design in Freight Transportation. *European Journal of Operational Research*, 122: 272-288.

Hane, C. A., Barnhart C., Johnson, E. L., Marsten R. E., Nemhauser G. L., Sigismondi G., 1995. "The Fleet Assignment Problem - Solving a Large-Scale Integer-Program," *Mathematical Programming*, 70(2), 211-232.

Ioachim, I., Desrosiers, J., Soumis, F. and Belanger, N., 1999. Fleet Assignment and Routing With Schedule Synchronization Constraints. *European Journal of Operational Research*, 119(1): 75-90.

Jarrah, A.I., Goodstein, J. and Narasimhan, R., 2000. An Efficient Airline Re-Fleeting Model for the Incremental Modification of Planned Fleet Assignments. *Transportation Science*, 34(4): 349-363.

Jarrah, A.I. and Strehler, J.C., 2000. An Optimization Model for Assigning Through Flights. *IIE Transactions*, 32(3): 237-244.

Jones, K.L., Lustig, I.J., Farvolden, J.M. and Powell, W.B., 1993. Multi-Commodity Network Flows: The Impact Of Formulation On De-Composition. *Mathematical Programming*, 62(1): 95-117.

Kabbani, N.M. and Patty, B.W., 1992. Aircraft Routing at American Airlines. *Proceedings of the 32nd Annual Symposium of AGIFORS*, Budapest, Hungary.

Kim, D., Barnhart, C., Ware, K. and Reinhardt, G., 1999. Multimodal Express Package Delivery: A Service Network Design Application. *Transportation Science*, 33(4): 391-407.

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

Magnanti, T.L. and Wong, R.T., 1981. Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. *Operations Research*, 29(3): 464-484.

Magnanti, T.L. and Wong, R.T., 1990. Decomposition Methods for Facility Location Problems. In: P.B. Mirchandani and R.L. Francis (Editors), *Discrete Location Theory*. John Wiley & Sons, Inc., New York, pp. 209-262.

Rexing, B., Barnhart, C., Kniker, T., Jarrah, A. and Krishnamurthy, N., 2000. Airline Fleet Assignment with Time Windows. *Transportation Science*, 34(1): 1-20.

Rushmeier, R.A. and Kontogiorgis, S.A., 1997. Advances in the Optimization of Airline Fleet Assignment. *Transportation Science*, 31(2): 159-169.

Subramanian, R., Scheff, R.P., Quillinan, J.D., Wiper, D.S. and Marsten, R.E., 1994. Coldstart - Fleet Assignment at Delta-Airlines. *Interfaces*, 24(1): 104-120.

Talluri, K.T., 1996. Swapping Applications in A Daily Airline Fleet Assignment. *Transportation Science*, 30(3): 237-248.