Composite Variable Formulations for Express Shipment Service Network Design

by

Andrew P. Armacost

B.S., Northwestern University (1989)
S.M., Massachusetts Institute of Technology (1995)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2000

Signature of Author ....
Department of Electrical Engineering and Computer Science
1 August 2000

Certified by ....
Cynthia Barnhart
Associate Professor, Civil and Environmental Engineering and Co-Director, Operations Research Center
Thesis Supervisor

Accepted by ....
James B. Orlin
Co-Director, Operations Research Center

Composite Variable Formulations for Express Shipment Service Network Design

by

Andrew P. Armacost

Submitted to the Department of Electrical Engineering and Computer Science
on 1 August 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In this thesis, we consider large-scale network design problems, specifically the problem of designing the air network of an express shipment (i.e., overnight) delivery operation. We focus on simultaneously determining the route structure, the assignment of fleet types to routes, and the flow of packages on aircraft. Traditional formulations for network design involve modeling both flow decisions and design decisions explicitly. The bounds provided by their linear programming relaxations are often weak. Common solution strategies strengthen the bounds by adding cuts, but the shear size of the express shipment problem results in models that are intractable.

To overcome this shortcoming, we introduce a new modeling approach that 1) removes the flow variables as explicit decisions and embeds them within the design variables and 2) combines the design variables into *composite variables*, which represent the selection of multiple aircraft routes that cover the demands for some subset of commodities. The resulting *composite variable formulation* provides tighter bounds and enables very good solutions to be found quickly. We apply this type of formulation to the express shipment operations of the United Parcel Service (UPS). Compared with existing plans, the model produces a solution that reduces the number of required aircraft by almost 11 percent and total annual cost by almost 25 percent. This translates to potential annual savings in the hundreds of millions of dollars.

We establish the composite variable formulation to be at least as strong as the traditional network design formulation, even when the latter is strengthened by Chvátal-Gomory rounding, and we demonstrate cases when strength is strictly improved. We also place the composite variable formulation in a more general setting by presenting it as a Dantzig-Wolfe decomposition of the traditional (intractable) network design formulation and by comparing composite variables to Chvátal-Gomory cuts in the dual of a related formulation. Finally, we present a composite variable formulation for the Pure Fixed Charge Transportation Problem to highlight the potential application of this approach to general network design and fixed-charge problems.

Thesis Supervisor: Cynthia Barnhart
Title: Associate Professor, Civil and Environmental Engineering and Co-Director, Operations Research Center

Contents

List of Figures

List of Tables

Acknowledgments

This thesis would not have been possible without the help of quite a few people. First, and foremost, I thank my wife, Kathy, for her unwavering love and support. She has been the stabilizing force in my life, a wonderful wife, and a great mother. To Ava and Audrey: Daddy is finally done with his "big paper!" To Mom, to Dad and Julia, to Bob and Mindy, and to Katie: thanks for being such a great part of my life and giving me the love and support I needed to succeed here and elsewhere.

My sincere gratitude goes to my advisor, Cindy Barnhart. It was her encouragement that led to my return to MIT. Her positive attitude is contagious and her ability to do the "heavy lifting" is unmatched. I have learned from her a great deal about research, problem solving, and life.

My thanks go to my thesis committee members – Tom Magnanti, Georgia Perakis, and Bill Hall – for their support and advice during this process. I have learned a great deal from them and their contributions to this thesis.

I am grateful to the United Parcel Service for the support it has provided to my thesis and to MIT. Specifically, I would like to thank Keith Ware and Alysia Wilson of the UPS OR group for their infectious energy and desire to create something for the good of their company.

I must thank the students, faculty, and staff of the MIT Operations Research Center for their friendship and advice over the last three years. The names are too many to mention, but the warm, supportive attitude at the Center has made a stressful three years much more enjoyable. I'd like to specifically thank Amy Cohn and Amr Farahat for the time they spent reviewing portions of this thesis and for their insightful comments and suggestions, which have surely made this a better document.

I thank the U. S. Air Force and the Department of Management at the Air Force Academy for enabling me to pursue my degrees. I look forward to returning to the faculty and contributing in a substantial and meaningful way.

Finally, as an Air Force member, am I required to acknowledge that the views expressed in this article are mine and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

Chapter 1

Introduction

In 1999, the U. S. package delivery industry generated an estimated $52 billion in revenues[1]. The domestic air portion accounted for $18 billion, domestic ground for $19 billion, and international delivery for $15 billion. Among the industry players, the United Parcel Service (UPS) is the largest, generating domestic revenues of $21.6 billion, $7.2 billion of which were due to air deliveries. The largest air carrier is Federal Express, with $9.7 billion in revenue due to domestic air delivery. Additional players in the industry include DHL Worldwide Express, Airborne Express, and Emory Worldwide.

The growth of e-business has had a dramatic effect on the package delivery industry, but the impact on express shipment service has been minor. The enormous increase in both consumer and business-to-business on-line transactions will generate an estimated $4.3 billion of additional revenue for the transportation industry in 2002[2]. The primary beneficiaries of this new market segment have been package delivery companies and less-than-truckload carriers. Of the 1999 shipping revenues attributable to e-business, 55% were captured by UPS, 32% by the U. S. Postal Service, and 10% by Federal Express. Yet, the role of *express* service in this market segment has been minimal, with only 2% of on-line purchases specified for overnight delivery.

This thesis is centered on the air portion of express shipment networks. With carriers charging premium price points for overnight delivery, the express air system represents an overwhelming proportion of revenue in the air freight segment. In addition, the cost of operating

[1] Standard & Poor's Commercial Transportation Industry Survey, February, 2000

[2] Zona Research report, cited by Standard & Poors Industry Survey

an air network is staggering due to the huge infrastructure required to provide air service. Improving the design of these service networks will yield significant cost savings.

These network design problems, when formulated as combinatorial optimization problems, are among the most difficult to solve. This difficulty arises from the need to model both package flow variables and integral design variables (i.e., aircraft routes). The linear programming relaxations tend to select fractional aircraft routes, which result in solutions that provide poor approximations to the true optimal solution. Advances in the theory of solving network design problems are geared toward improving the approximation provided by the LP relaxation and have improved our ability to solve problems within this class. Unfortunately, the combination of the massive scale of the express shipment problem and the inherent difficulty of solving its mathematical representation render these advances ineffective for the problems we consider.

For that reason, we introduce a new approach for solving the express shipment service network design problem. The foundation of this approach is the use of *composite variables.* At their core, the composites capture package flows *implicitly*, meaning that package flow variables are no longer a part of the formulation. Furthermore, composites absorb a significant portion of the problem's inherent complexity that results from interactions between aircraft routes. The overall result is that the composites prevent many fractional solutions from ever appearing in the linear programming relaxation. Thus, a composite-based network design model is better approximated by its LP relaxation and, therefore, easier to solve.

1.1 Contributions

With the goal of developing and utilizing a practical solution methodology for network design, we make the following significant contributions in this thesis:

- Develop a **robust solution methodology** for solving the Express Shipment Service Network Design (**ESSND**) problem. Standard polyhedral methods for network design and network loading problems are not effective on instances of realistic size. The composite variable formulation provides stronger bounds along with the flexibility to handle practical constraints that make traditional formulations intractable. Computations with this model are fast, making it a useful tool to support network planners.

- Demonstrate the **practical significance** of the composite variable approach on a carrier-specific instance of the **ESSND** problem. This instance, which is representative of many others, could not be solved otherwise. We demonstrate the potential to save hundreds of millions of dollars in the annual cost of owning and operating aircraft.

- Establish the **theoretical foundation** for this method. We show the equivalence of the composite variable formulation with traditional models and show the composite variable formulation provides stronger bounds on the optimal integer solution.

- Demonstrate **how to generalize** the composite variable approach to a broader class of problems. We do this by relating composite variable formulations to Dantzig-Wolfe decomposition and relating the specific operation of creating a composite to the cutting plane methods of Chvátal and Gomory.

1.2 Thesis Overview

This structure of this thesis is designed to emphasize the development of the composite variable approach for solving large-scale, practical problems. Chapter 2 places the express shipment planning problem in the context of broader classes of problems, namely the Network Design Problem (**NDP**) and the Network Loading Problem (**NLP**). This chapter summarizes recent work on these broader classes of problems, as well as the techniques that have been applied to the Express Shipment Service Network Design (**ESSND**) problem.

In Chapter 3, we reformulate the **ESSND** problem using a composite variable formulation that we call the Aircraft Routing Model (**ARM**). We present several variations of **ARM** to illustrate methods for incorporating additional operational requirements. For the planning problem specific to UPS, we describe the algorithms used to generate the set of composite variables over which **ARM** is optimized.

Having presented the basis for the composite variable formulation and the procedures for building the model, in Chapter 4 we apply this modeling approach to an actual UPS planning problem. We show that the manner in which we build composite variables affects both run-time and solution quality. We explore the trade-off between the amount of additional work (running time) and the marginal benefit when we alter the set of composite variables. We then compare

ARM's solution directly with the UPS planners' solution. **ARM** builds solutions with tens of millions of dollars less in annual operating cost and hundreds of millions of dollars less in annual ownership cost. Finally, we highlight the flexibility of this modeling approach by demonstrating the ease of exploring additional scenarios and the ease of incorporating additional operating requirements in the composite variables.

With the practical impact of composite variable formulations firmly established, Chapter 5 presents the theoretical foundation for this method and proves its LP relaxation to be stronger than that of traditional network design formulations. We accomplish this by showing a transition from the original formulation to the composite variable formulation via a third "intermediate" model. This discussion justifies (from the theoretical perspective) the implemented version of **ARM** described in Chapters 3 and 4 by showing that, under reasonable operating assumptions, the implemented version of **ARM** yields the optimal solution to the original **ESSND** problem.

Finally, in Chapter 6 we present **ARM** in a more general setting. We provide an interpretation of **ARM** as a Dantzig-Wolfe decomposition of the **ESSND** formulation. Through this decomposition framework, we are able to readily derive bounds on the optimal **ESSND** solution, providing an alternative to the weak bounds given by the **ESSND** LP relaxation. In addition, we link the process of building composites to the well-known cutting plane methods of Chvátal and Gomory. Finally, we take the initial step of applying this modeling technique to a broader class of network design problems by constructing a composite variable formulation for the Pure Fixed Charge Transportation Problem (**PFCTP**).

The final chapter summarizes the results and contributions of this thesis. Equally important is the identification of future areas of research. This new formulation approach has both practical and theoretical significance. It represents a significantly different approach for solving network design problems and, potentially, other types of integer and mixed integer programming problems.

Chapter 2

Network Design and Express Shipment Service

Network design encompasses a wide range of planning problems encountered in transportation, telecommunications, manufacturing, and other areas. Whether involving the construction of physical networks or the determination of services to be provided, problems of this form arise in all levels of planning – from strategic out-year planning to real-time operations and control. The core idea is the same: we find the best assignment of capacity to the arcs in the network and of commodity flows on those arcs. In this chapter, we define the basic Network Design Problem (**NDP**) and a variant known as the Network Loading Problem (**NLP**). We then extend **NLP** to the problem of express (i.e., overnight) package delivery service.

2.1 Network Design Problems

Network design problems involve *design* choices, which are discrete, and *flow* choices, which are typically continuous. We are given a directed graph, $G = (N, A)$, and a set of commodities, K, specified by origin-destination pairs. Let c_{ij}^k be the (linear) cost per unit of commodity $k \in K$ flown on arc $(i,j) \in A$ and let d_{ij} be the fixed cost of using arc (i,j). The Network Design Problem (**NDP**), as described in Magnanti and Wong [60] and Ahuja et al. [2], is:

$$\min \sum_{k\in K} \sum_{(i,j)\in A} c_{ij}^k x_{ij}^k + \sum_{(i,j)\in A} d_{ij} y_{ij}$$

subject to:

$$\sum_{k\in K} x_{ij}^k \leq u_{ij} y_{ij} \quad (i,j)\in A \tag{2.1}$$

$$\sum_{j:(i,j)\in A} x_{ij}^k - \sum_{j:(j,i)\in A} x_{ji}^k = \begin{cases} b^k & \text{if } i = O(k) \\ -b^k & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \quad i\in N,\ k\in K \tag{2.2}$$

$$x_{ij}^k \geq 0 \quad (i,j)\in A,\ k\in K \tag{2.3}$$

$$y_{ij} \in \{0,1\} \quad (i,j)\in A. \tag{2.4}$$

The *forcing* constraints (2.1) ensure that the flow on any arc does not exceed the capacity assigned to that arc. Constraints (2.2) ensure *conservation of flow* for the commodities. Finally, the flows are nonnegative (2.3) and the design variables are binary (2.4). In the uncapacitated version, each arc capacity, u_{ij}, is no smaller than the sum of all commodity demands.

The Network Loading Problem (**NLP**), as presented by Magnanti and Mirchandani [56], is a variant of the Network Design Problem in which types of capacity, known as *facilities*, may be assigned in integer quantities to the network arcs. The **NLP** does not include flow costs. Let F denote the set of facility types, let d_{ij}^f represent the cost of installing one unit of facility type f on arc (i,j), let u_{ij}^f denote the capacity provided by one unit of facility f on arc (i,j), and let y_{ij}^f be the variable corresponding to the decision of how many units of f to assign to arc (i,j). The **NLP** is defined as follows:

$$\min \sum_{f\in F} \sum_{(i,j)\in A} d_{ij}^f y_{ij}^f$$

subject to:

$$\sum_{k\in K} x_{ij}^k \leq \sum_{f\in F} u_{ij}^f y_{ij}^f \quad (i,j)\in A \tag{2.5}$$

$$\sum_{j:(i,j)\in A} x_{ij}^k - \sum_{j:(j,i)\in A} x_{ji}^k = \begin{cases} b^k & \text{if } i=O(k) \\ -b^k & \text{if } i=D(k) \\ 0 & \text{otherwise} \end{cases} \quad i\in N,\ k\in K \tag{2.6}$$

$$x_{ij}^k \geq 0 \quad (i,j)\in A,\ k\in K \tag{2.7}$$

$$y_{ij}^f \in \mathbb{Z}_+ \quad (i,j)\in A,\ f\in F. \tag{2.8}$$

Magnanti and Wong [60], Minoux [61], and Kim et al. [52] provide surveys of network design models and applications. Magnanti and Wong [60] provide a unified framework for describing network design problems and deriving network design algorithms. They also demonstrate the wide range of combinatorial problems that are specializations or variations of network design, highlighting the broad impact and potential application of network design models and solution strategies.

Characterizing polyhedra and deriving valid inequalities for network design can be traced to the development of valid inequalities for 0-1 programming (Wolsey [72] and Crowder et al. [27]) and the development of valid inequalities for fixed-charge network problems (Van Roy and Wolsey [70] and Padberg et al. [66]). Magnanti et al. [57] characterize the convex hull of Network Loading Problems that involve multiple commodities and a single facility type. Magnanti and Mirchandani [56] study the single-commodity, multi-facility network loading problem and show how to characterize the optimal solution of some two- and three-facility problems by a linear program. Pochet and Wolsey [67] investigate polyhedral properties of single-arc multi-facility network design problems, where the facility capacities are integer multiples of some base capacity unit. Magnanti et al. [58] model the two-facility capacitated network loading problem, for which they describe three types of valid inequalities and demonstrate the effectiveness of these inequalities in tightening the LP relaxation. They generalize the results to multi-facility problems, where facility capacities are integer multiples of some base capacity. Bienstock and Günlük [22] extend two of these valid inequalities and embed them in a cutting plane algorithm. Chopra et al. [24] derive additional inequalities for the case of the single-commodity, two-facility network design problem. Bienstock et al. [21] compare formulations for the single-facility multicommodity network design problem, describe two classes of valid inequalities, and characterize the corresponding polyhedron for a three-node graph. Atamtürk [4] extends many of these polyhedral results to the case of multiple commodities and multiple facilities with *arbitrary* capacities.

Development of algorithms that embed polyhedral elements include Bienstock and Günlük [23], who describe a cutting plane algorithm for the problem of network design to minimize the maximum load on any arc. Barahona [10] solves both the bifurcated and nonbifurcated versions of the network loading problem. Günlük [41] demonstrates effective use of strong cuts within a branch-and-cut framework that uses a knapsack branching rule. Stallaert [69] describes a simple procedure to derive network inequalities for capacitated fixed charge network problems by exploiting properties of fractional extreme point solutions to the LP relaxation.

Balakrishnan et al. [6] study the Two-Level Network Design Problem (**TLNDP**), looking at relationships between formulations of the undirected and directed versions of the **TLNDP**. Further, they develop heuristic algorithms and analyze their worst-case performance. For the Multi-level Network Design Problem (**MLNDP**), Balakrishnan et al. [5] present a solution methodology that performs design variable fixing based on structural properties of known optimal solutions and dual ascent to generate lower and upper bounds. Balakrishnan et al. [9] address local access network expansion planning for telecommunications companies, deriving valid inequalities based on the problem-specific polyhedral structure. They use the inequalities in a Dynamic Program (DP) to solve the uncapacitated version of the problem. The DP is embedded within a Lagrangian relaxation scheme and the method is shown to provide good lower and upper bounds. Balakrishnan et al. [7] present worst-case bounds for heuristics and LP relaxations of the overlay optimization problem and demonstrate worst-case bounds for the uncapacitated multicommodity network design problem. Balakrishnan et al. [8] introduce a multi-tier survivable network design problem for which they derive a solution procedure that solves the single-tier subproblems as matroids.

The development of network design heuristics with worst-case bounds begins with Goemans and Bertsimas [33], who develop two heuristics for the survivable network design problem, relying on a special property (called the parsimonious property) of a classical formulation's LP relaxation. Agrawal et al. [1] present the first approximation algorithm (i.e., polynomially solvable) for the general Steiner network problem. Goemans and Williamson [34] extend this approach by obtaining an approximation with a minimum weight perfect matching problem. Williamson et al. [71] present a primal-dual approach that is the first approximation algorithm for the more general survivable network design problem. Jain [47] presents a factor 2 approximation for the generalized Steiner Network problem using its linear programming relaxation and iteratively rounding-off the solution. Gabow et al. [31] improve the efficiency of the Williamson et al. algorithm and Hochbaum and Naor [43] extend it to network design problems with additional requirements. Bertsimas and Teo [20] describe a primal-dual framework to design and analyze integer programming approximation algorithms that are based on the construction of valid inequalities. Karger [49] presents random sampling-based approximation algorithms as a tool for solving undirected graph problems, which include network design problems.

Other examples of applying optimization techniques to network design problems include Magnanti et al. [59], who study the application of Benders decomposition to the uncapacitated network design problem. In addition to presenting new Benders cuts for this problem, they derive known valid inequalities as Benders cuts. They also demonstrate the effectiveness of variable elimination preprocessing and a dual ascent procedure to accelerate the decomposition algorithm. Alevras et al. [3] develop cutting plane and heuristic approaches for solving the problem of installing capacity on arcs in a telecommunications network and cite computational results using real-world data. Holmberg and Hellstrand [45] present a Lagrangian-based heuristic embedded within a branch-and-bound framework for solving the uncapacitated network design problem. Myung et al. [62] design survivable networks with a specified allowable loss. They develop an integer programming formulation solved by a heuristic procedure and apply it real-world problems. Gabrel et al. [32] derive exact procedures for solving multicommodity network flow problems with general step cost functions and use Benders decomposition as a solution procedure. This class of problem includes the multi-facility network loading problem as a special case.

Multicommodity network flow (**MCNF**) problems are at the core of network design problems. Ahuja et al. [2] present general techniques for solving **MCNF** problems, such as Lagrangian relaxation and Dantzig-Wolfe decomposition. Barnhart [11] develops dual ascent procedures for solving large-scale **MCNF** problems. Farvolden et al. [30] solve the **MCNF** problem using primal partitioning and Dantzig-Wolfe decomposition. Barnhart and Sheffi [18] develop primal-dual heuristics for **MCNFs**. Barnhart et al. [13] solve large-scale **MCNF** problems with column generation methods and Barnhart et al. [14] use branch-and-price to solve large-scale *integer* **MCNF** problems. Barnhart et al. [15] use branch-and-price-and-cut to solve integer **MCNF** problems in which the flow of each commodity is constrained to a single path between the commodity's origin and destination. Jones et al. [48] demonstrate the effect of formulation strategy on the solution of multicommodity flow problems. Kim and Barnhart [51] and Krishnan et al. [53] explore these strategies in the context of express shipment service network design. Leighton et al. [55] develop approximation algorithms for **MCNF** problems.

A common extension of network design, particularly in transportation applications, involves additional restrictions on the design variables. These often stem from the need to represent transportation networks dynamically and to enforce a flow of the *design* components. Such constraints may take the following form:

$$\sum_{j:(i,j)\in A} y_{ij}^f - \sum_{j:(j,i)\in A} y_{ji}^f = 0 \quad i \in N,\ f \in F, \tag{2.9}$$

which ensures conservation of flow for the *design* variables. A general scheme for defining service network design is proposed by Crainic [26], who surveys and classifies service network design and general network design problems and formulations.

For express shipment service network design, Barnhart and Schneur [17] address the problem of designing a single-hub overnight delivery network using column generation techniques to obtain near-optimal solutions. Kim et al. [52] apply branch-and-price-and-cut methods to the multi-hub express shipment problem using a heuristic solution strategy. Grünert and Sebastian [39] identify planning tasks faced by postal and express shipment companies and define corresponding optimization models.

Farvolden and Powell [29] use subgradient-based heuristics for service network design in the motor carrier industry. They use duals from multicommodity flow problems to drive the selection of services using local search. In railroad planning, Gorman [36] and [37] demonstrates the use of tabu search and genetic algorithms for the design of freight railway operating service.

Newton et al. [65] model the railroad blocking plan problem as a network design problem and generate solutions using branch-and-price. Ziarati et al. [73] solve the locomotive assignment problem using Dantzig-Wolfe decomposition with shortest path subproblems. Ziarati et al. [74] solve the locomotive assignment problem using a branch-and-cut approach.

In the commercial airline industry, there has been no application of network design models to the problem of determining aircraft routes and service schedules. Lederer and Nambimadom [54] characterize the elements that influence the quality and reliability of the network design of commercial airlines. The use of optimization methods for the airlines has been well-documented. Applications include fleet assignment (see Rexing et al. [68] and Barnhart et al. [12]), crew scheduling (see Barnhart and Shenoi [19] and Hoffman and Padberg [44]), and aircraft maintenance routing (see Gopalan and Talluri [35]).

2.2 Express Shipment Service Network Design

In this section, we extend **NLP** to express shipment operations. The Express Shipment Service Network Design (**ESSND**) problem is characterized by a certain structure of the underlying network and by additional constraints placed on the design elements. This section describes the operations of a typical express shipment carrier in order to motivate the problem formulation. We defer carrier-specific details to later sections.

2.2.1 Problem Description

Express shipment carriers operate systems of aircraft, trucks, sorting facilities, equipment, and personnel to move packages overnight between customers. While they use this set of resources during the day to move its non-express packages, the problem we consider involves only overnight operations. In response to demand projections and operational restrictions, the carriers must determine which routes to fly, which fleet types to assign to those routes, and how to assign packages to those aircraft. We refer to the resulting plan, along with the resources to operate it, as the *Next-Day Air (NDA) network.*

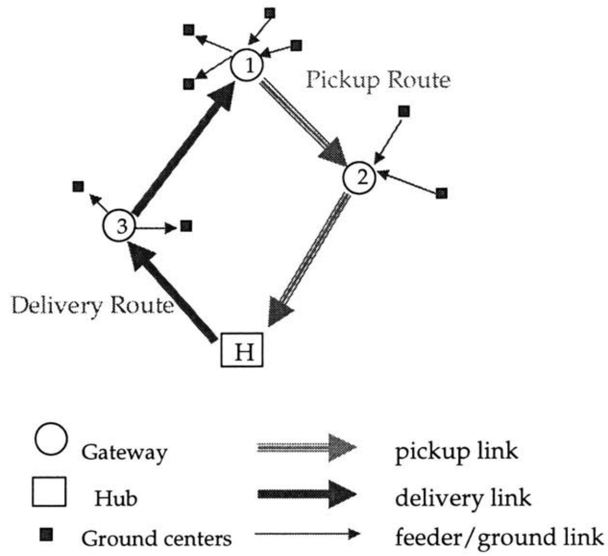

Figure 2-1: Example Next-Day Air (NDA) routes

The Physical System

The NDA system consists of *gateway* locations, which serve as points at which packages enter (or exit) the air network; *hub* locations, where packages are sorted; and aircraft of multiple fleet types. Consider the network shown in Figure 2-1. Packages arrive from customer *centers* to gateways either on trucks or on small aircraft that service remote locations. When packages enter the air system through a gateway (e.g., node 1), they are loaded onto an aircraft and are transported to a hub (e.g., node H) not later than the hub's *sort start time*. Flying to the hub is done by direct routes or, as in Figure 2-1, via an intermediate gateway (e.g., node 2).

Upon arrival at the hub, packages are unloaded from the aircraft, sorted, and loaded onto aircraft for delivery to their destination gateway. During the sorting process, the inbound planes remain at the hub until they are loaded and ready to start their delivery routes. Hubs may also serve as gateways since packages may either originate or terminate at these locations.

On delivery routes, planes can depart the hub no earlier than the *sort end time*. They deliver packages to the gateway locations, which then sort the packages and send them to ground sorting facilities via truck or feeder aircraft. From there, the packages are delivered to customers.

The aircraft inventory consists of multiple aircraft types. Each aircraft type has operating characteristics that influence which routes it can fly. These include maximum flying range, effective speed, restrictions on the locations at which it can land, and package capacity. In addition to the large jet aircraft, a fleet of small aircraft provides a flexible source of capacity to handle excess demands that arise during the actual operation of the air network.

Demands

Demands are specified by gateway origin, gateway destination, and total volume between the two. While actual demands are customer-to-customer, demand estimates are assumed to be compressed into center-to-center demands and then into gateway-to-gateway demands. Thus, when overnight delivery may be made entirely on ground vehicles, this demand will not enter the NDA network.

The units for measuring demand may vary. The granularity of the units influences one's choice of model(s) since fractional flows may be acceptable using one measurement but not another. We consider both packages and containers. Aircraft can typically hold thousands or tens-of-thousands of packages and only tens of containers. Finally, these demands are input to the model as deterministic figures. Whether they are expected values, conservative estimates, or part of a profile or distribution of demand estimates is external to our planning problem.

Route Restrictions

To ensure appropriate customer service levels, boundaries are set for pickup and delivery. For each gateway location, the carrier assigns *level-of-service* (LOS) requirements in the form of an *Earliest Pickup Time* (EPT), which specifies the earliest time an aircraft can depart from that location, and a *Latest Delivery Time* (LDT), which specifies the latest time at which packages can be delivered to the gateway. Timing requirements at hubs are designated by *sort start times* and *sort end times*. Sort start represents the latest time at which planes can arrive to the hub on a pickup route and have their packages sorted and loaded onto delivery routes. Sort end represents the latest time at which packages may be loaded onto outbound aircraft and, therefore, the earliest time at which planes may depart on delivery routes. Restrictions are placed on where a plane type can land, which is influenced by factors such as runway length, physical space on the ramp, and noise restrictions at airports. Finally, there are limits on the number of legs a plane can fly on a pickup or delivery route.

Cost Elements

Cost is incurred both for utilizing aircraft and for handling packages. Each aircraft incurs three types of cost. First, *variable operating cost* is based on block hours flown (i.e., flying time plus taxi time). Second, a fixed *cycle cost* is incurred on each flight leg. Third, *ownership cost* is the daily cost of owning the aircraft. Package flow (shipment) cost has two components: a cost based on block time and a fixed handling cost. The package cost elements are typically much smaller than the cost of owning and operating the aircraft.

2.2.2 Formulations

The planning problem faced by express shipment carriers is characterized as follows. We seek to minimize cost by simultaneously selecting routes, aircraft types for each route, and package flows through the network. Additional constraints, on top of aircraft balance specified in (2.9), include:

- enforce the sorting capacity at each hub, $e_h, \ h \in H$
- limit the number of utilized aircraft (of each fleet type) to the number available, $n_f, \ f \in F$
- limit the number of aircraft landing at each hub to the hub's landing capacity, $a_h, \ h \in H$
- satisfy level-of-service (LOS) requirements for pickup and delivery
- arrive to and depart from hub locations according to the sort start and end times.

We first define the problem notation and present a node-arc formulation that essentially extends **NLP** to the express shipment context. We then describe two methods for decomposing the **ESSND** model to yield formulations with fewer constraints. The first decomposition represents package flow variables by origin-destination (O-D) path flows. The second collects O-D commodities by origin and yields a model whose extreme point solutions are *sets of path flows* rooted at a common origin.

Let $G = (N, A)$ be the network of nodes and arcs on which we are creating the service network. Using *path-based* variables for the aircraft routes, we partially enforce aircraft balance constraints (2.9) through this route-based variable definition. Let R^f be the set of routes that can be flown by fleet type $f \in F$. Define the integer decision variable y_r^f to be the number of times we fly route $r \in R^f$ with fleet type $f \in F$. The cost of this aircraft route is denoted by d_r^f, which is simply the cost of flying each arc in route r with fleet type f. We map each aircraft route(f, r) to the arcs in A with the indicator δ_{ij}^{fr}, which equals 1 when flight arc (i, j) is contained in aircraft route (f, r) and 0 otherwise. We map the arc corresponding the sort at hub h with the indicator δ_{ij}^h, and we map each route at each hub prior to the sort with the indicator δ_h^r. Associated with the start and end of each route is the indicator β_i^r, which equals 1 when i is the route's origin, -1 when i is the route's destination, and 0 otherwise. The **ESSND** formulation, introduced in Kim et al. [52], is given by:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k + \sum_{f \in F} \sum_{r \in R^f} d_r^f y_r^f$$

subject to:

$$\sum_{k \in K} x_{ij}^k \leq \sum_{f \in F} \sum_{r \in R^f} \delta_{ij}^{fr} u_r^f y_r^f \quad (i,j) \in A \tag{2.10}$$

$$\sum_{j:(i,j) \in A} x_{ij}^k - \sum_{j:(j,i) \in A} x_{ji}^k = \begin{cases} b^k & \text{if } i = O(k) \\ -b^k & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \quad i \in N,\ k \in K \tag{2.11}$$

$$\sum_{r \in R^f} \beta_i^r y_r^f = 0 \quad i \in N,\ f \in F \tag{2.12}$$

$$\sum_{k \in K} \sum_{(i,j) \in A} \delta_{ij}^h x_{ij}^k \leq e_h \quad h \in H \tag{2.13}$$

$$\sum_{r \in R^f} y_r^f \leq n_f \quad f \in F \tag{2.14}$$

$$\sum_{f \in F} \sum_{r \in R^f} \delta_h^r y_r^f \leq a_h \quad h \in H \tag{2.15}$$

$$x_{ij}^k \geq 0 \quad (i,j) \in A,\ k \in K \tag{2.16}$$

$$y_r^f \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F. \tag{2.17}$$

Constraints (2.10)-(2.11) are the same as in **NLP**, with the aircraft movements (i.e., the design variables) modeled as path flows versus arc flows. Constraints (2.12) are the path-based form of the *aircraft balance* constraints, whose nonzero entries correspond to the origin and destination of each aircraft route. Additional constraints enforce *sort capacities* at the hubs (2.13), *number of available aircraft* of each fleet type (2.14), and *landing capacities* at the hubs (2.15).

Decomposition Strategy

The huge number of conservation of flow constraints (2.11) yields an intractable model for realistic problem instances. To reduce the number of constraints, Kim et al. [52] apply Dantzig-Wolfe decomposition (see Dantzig and Wolfe [28]) with respect to the package flow variables. The master problem and subproblem structures depend upon how we define the *commodities.* The first definition is in the sense described earlier – commodities defined by *origin-destination* pairs. The second definition groups O-D commodities by origin location into *supercommodities.* Our presentation of the decomposition is initially in terms of a *generic* commodity set, $\mathcal{K}$, and we later explore the effect of defining commodities either by origin-destination pair or by origin.

Assume we are given *fixed* aircraft routes, $\overline{\mathbf{y}}$, that satisfy plane count constraints and landing capacities at the hubs. Let $\overline{\mathbf{u}}$ be the vector of arc capacities that result from these routes and let $\mathbf{c}^k$ be the vector of arc costs for commodity k. (Node capacities, such as those specified in constraints (2.13), can be transformed into arc capacities by splitting the node and adding a directed arc with capacity e_h between the nodes.) Thus, for fixed aircraft routes, the resulting multicommodity network flow problem becomes:

$$\min \sum_{k \in \mathcal{K}} \left(\mathbf{c}^k\right)' \mathbf{x}^k \tag{2.18}$$

subject to:

$$\sum_{k \in \mathcal{K}} \mathbf{x}^k \leq \overline{\mathbf{u}} \tag{2.19}$$

$$\mathcal{N}^k \mathbf{x}^k = \mathbf{b}^k \quad k \in \mathcal{K} \tag{2.20}$$

$$x_{ij}^k \geq 0 \quad (i,j) \in A. \tag{2.21}$$

In this model, $\mathcal{N}^k$ is the node-arc incidence matrix for commodity k (i.e., with components given in (2.11)) and $\mathbf{b}^k$ is its vector of demands.

Let $\mathcal{E}^k$ be the set of extreme points corresponding to the k^{th} commodity's network flow constraints in (2.11). The general form of the *master problem* is given by:

$$\min \sum_{k \in \mathcal{K}} \sum_{e \in \mathcal{E}^k} \left[\left(\mathbf{c}^k \right)' \mathbf{x}_e^k \right] \lambda_e^k \tag{2.22}$$

subject to:

$$\sum_{k \in \mathcal{K}} \sum_{e \in \mathcal{E}^k} \mathbf{x}_e^k \lambda_e^k \leq \overline{\mathbf{u}} \tag{2.23}$$

$$\sum_{e \in \mathcal{E}^k} \lambda_e^k = 1 \quad k \in \mathcal{K} \tag{2.24}$$

$$\lambda_e^k \geq 0 \quad e \in \mathcal{E}^k,\ k \in \mathcal{K}. \tag{2.25}$$

We can work with a *restricted master problem* and generate additional columns as needed. Letting $\boldsymbol{\pi}$ be the duals associated with constraints (2.23), the k^{th} subproblem to generate new columns corresponding to commodity k is given by:

$$\min \left[\mathbf{c}^k - \boldsymbol{\pi} \right]' \mathbf{x}^k \tag{2.26}$$

subject to:

$$\mathcal{N}^k \mathbf{x}^k = \mathbf{b}^k \tag{2.27}$$

$$x_{ij}^k \geq 0 \quad (i,j) \in A. \tag{2.28}$$

Let σ^k denote the dual associated with the k^{th} convexity constraint (2.24). If the subproblem's objective value is less than σ^k, its solution is an extreme point with negative reduced cost and is added to the restricted master problem. How we solve subproblem (2.26)-(2.28) depends upon how we define the commodity set, $\mathcal{K}$.

Origin-Destination Formulation. We first present the case when the commodities are defined by *origin-destination* pairs, that is $\mathcal{K} \equiv K$. The elements of the demand vector, $\mathbf{b}^k$, are given by:

$$b_i^k = \begin{cases} b^k & \text{if } i = O(k) \\ -b^k & \text{if } i = D(k) \\ 0 & \text{otherwise,} \end{cases}$$

where b^k is the total volume for commodity k with origin $O(k)$ and destination $D(k)$. The extreme points of the k^{th} subproblem defined in (2.26)-(2.28) are *paths* from $O(k)$ to $D(k)$. Denote the set of extreme points for the k^{th} subproblem by $\mathbf{x}_p^k$, $p \in P^k$. Solving the k^{th} subproblem yields the **shortest path,** p^*, from $O(k)$ to $D(k)$ carrying b^k units of flow. We represent the solution by first defining the indicator $\delta_{ij}^p = 1$ if path p^* includes arc $(i,j) \in A$, and 0 otherwise. Using $\boldsymbol{\delta}^{p^*}$ to denote the vector of indicators, the solution of arc flows is given by $\mathbf{x}_{p^*}^k = \boldsymbol{\delta}^{p^*} b^k$, or a flow on each arc of $\delta_{ij}^{p^*} b^k$. The cost of the extreme point is $\left(\mathbf{c}^k\right)' \mathbf{x}_{p^*}^k = \left(\mathbf{c}^k\right)' \boldsymbol{\delta}^{p^*} b^k$.

Let **ESSND-OD** be the formulation of **ESSND** that decomposes package flows by origin-destination pairs, as just described. Each shortest path subproblem can be solved efficiently (see Ahuja et al. [2] for examples). The number of shortest path subproblems equals the number of O-D commodities. The master problem has a convexity constraint for each subproblem (2.24). While the number of subproblems might be in the thousands, this is several orders of magnitude smaller than the number of flow conservation constraints in the original **ESSND** formulation.

Origin Formulation. We next consider the decomposition that arises when the elements of commodity set $\mathcal{K}$ are defined by their *origin* location. We group all O-D commodities that share

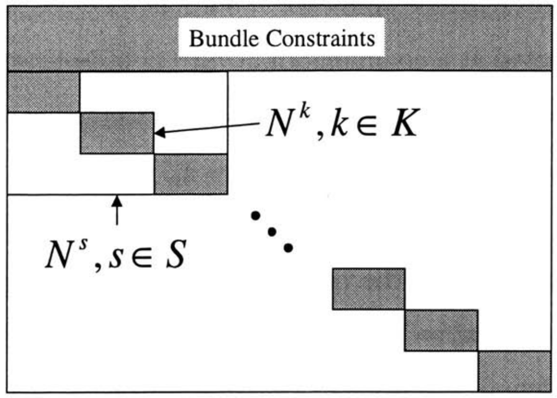

Figure 2-2: Matrix structure for multi-commodity network flow problem on fixed aircraft routes

a common origin into a single *supercommodity*. Let S denote the set of supercommodities, which has the same cardinality as the set of origin locations. Let K^s be the set of O-D commodities constituting $s \in S$. Each O-D commodity is contained in *exactly one* supercommodity.

Assume (without loss of generality) that a given supercommodity, s, comprises the first $|K^s|$ O-D commodities. Define the demand vector $(\mathbf{b}^s)' = \left[(\mathbf{b}^1)' \, (\mathbf{b}^2)' \ldots (\mathbf{b}^{|K^s|})'\right]$, let $(\mathbf{c}^s)' = \left[(\mathbf{c}^1)' \, (\mathbf{c}^2)' \ldots (\mathbf{c}^{|K^s|})'\right]$, and let $\mathcal{N}^s$ be the block diagonal matrix with blocks $\mathcal{N}^1 \ldots \mathcal{N}^{|K^s|}$ (see Figure 2-2). For each supercommodity $s \in S$, an extreme point is found by solving the subproblem:

$$\min \; [\mathbf{c}^s - \boldsymbol{\pi}]' \, \mathbf{x}^s \tag{2.29}$$

subject to:

$$\mathcal{N}^s \mathbf{x}^s = \mathbf{b}^s \tag{2.30}$$

$$x_{ij}^s \geq 0 \quad \forall (i,j) \in A. \tag{2.31}$$

Notice that each of the $|S|$ subproblems separates by its component origin-destination commodities. $\mathcal{N}^s$ separates by $k \in K^s$ and the k^{th} "sub-subproblem" has arc costs $\mathbf{c}^k - \boldsymbol{\pi}$. Thus, generating the extreme point is accomplished by solving the $|K^s|$ separable shortest path problems.

We denote the set of extreme points for subproblem s by $\mathbf{x}_q^s$, $q \in Q^s$. Each extreme point is a collection of paths rooted at a common origin with flow destined for $|K^s|$ different destinations. Let q^k be the path in extreme point q corresponding to the flow of commodity $k \in K^s$. Let $\boldsymbol{\delta}^{qk}$ be the vector of indicators for this path, where $\delta_{ij}^{qk} = 1$ if arc (i,j) is in the path and $\delta_{ij}^{qk} = 0$ otherwise. Each arc on this path has a flow of b^k units of commodity k (and possibly additional flow of another commodity) and the vector of arc flows is given by $\boldsymbol{\delta}^{qk} b^k$. The complete extreme point solution is given by $\mathbf{x}_q^s = \sum_{k \in K^s} \boldsymbol{\delta}^{qk} b^k$ with cost $c_q^s = (\mathbf{c}^s)' \mathbf{x}_q^s$.

Let **ESSND-O** denote the network design formulation that decomposes package flows according to origin-based supercommodities, as just described. This differs from **ESSND-OD** in that a) the restricted master problem is smaller in **ESSND-O** because the number of convexity constraints equals the number of origin gateways; and b) each path generated in **ESSND-OD** is represented in the master problem with its own convexity constraint multiplier while a path generated in **ESSND-O** is grouped with the other paths in q by a common convexity constraint multiplier. Thus, in the latter formulation, to re-use a path that had been previously generated requires that it be re-generated as part of a different extreme point (i.e., a different set of paths).

The number of shortest paths found at each iteration of the decomposition algorithm equals the number of origin-destination commodities, regardless of the decomposition strategy (i.e., either O-D commodity-based or supercommodity-based). Under certain conditions, the **ESSND-O** subproblems can be solved more efficiently. Consider the case when, for a particular $s \in S$, neither arc costs nor the node-arc incidence matrices vary by $k \in K^s$. That is, all origin-destination commodities in the supercommodity have the same vector of arc costs and the same network on which to flow. This yields a single cost vector $\mathbf{c}^s$ of length $|A|$ and a single node-arc incidence matrix $\mathcal{N}^s$ of dimension $|N| \times |A|$. We construct the common demand vector, $\mathbf{b}^s$, of length $|N|$, with components as follows:

$$b_i^s = \begin{cases} \sum\limits_{k \in K^s} b^k & \text{if } i = O(s) \\ -b^k & \text{if } i = D(k), \text{ for } k \in K^s \\ 0 & \text{otherwise.} \end{cases}$$

$O(s)$ is the common origin for the commodities in K^s. A solution to this problem is a *tree*, with the path for each $k \in K^s$ (from $O(k)$ to $D(k)$) carrying b^k units of flow. Generating this minimum length tree has the same *worst-case* complexity as finding a *single* shortest path and may be accomplished by solving a modified version of a standard shortest path algorithm such as Dijkstra's algorithm (see Ahuja et al. [2]). Thus, these assumptions on the cost structure reduce the work required to solve the **ESSND-O** subproblems.

Finally, we formalize the relationship between these two decomposition strategies.

Theorem 1 ***ESSND-O*** *and* ***ESSND-OD*** *(and their LP relaxations) are equivalent formulations.*

Proof. For a fixed aircraft route solution (either fractional or integral), **ESSND-OD** and **ESSND-O** are decompositions of the *same* multicommodity flow problem defined in (2.18)-(2.21) and yield solutions of the same cost. This, combined with the fact that both formulations consist of identical aircraft route variables and aircraft constraints, yields the desired result. ■

2.3 Carrier-Specific Model

In the development of the **ESSND** formulations, we made no assumptions about the route structure and the underlying time-space network. For the case of a particular express shipment carrier (UPS), we impose the additional assumptions:

Assumption 1. Package flow costs are zero.

Assumption 2. Gateway-hub demands are given in lieu of gateway-gateway demands.

Carrier-specific operations affect the underlying time-space network on which we design our service network. In this section, we summarize the construction process for the time-space network and we present a version of **ESSND** that takes advantage of this network structure. Details of these procedures are found in Kim et al. [52], and Kim [50].

2.3.1 Time-Space Network Construction

We exploit the carrier-specific requirement that each pickup route ends at a hub, each delivery route begins at a hub, and the number of legs on a pickup or a delivery route cannot exceed

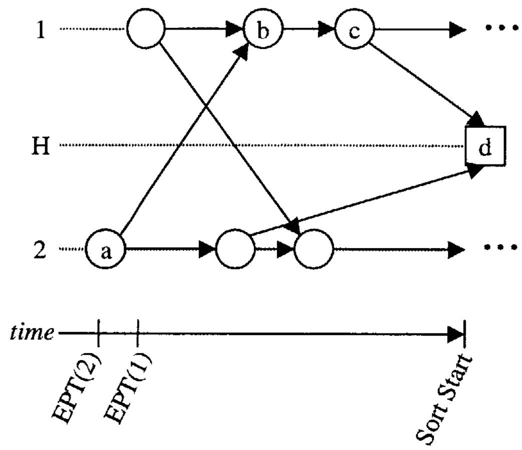

Figure 2-3: Derived time-space network for pickup routes of a single fleet type

two. For each fleet type, we construct a time-space network (Figure 2-3, which shows the pickup side of a simple operation). This figure shows two gateways and a single hub for which we must represent aircraft movements in both space and time. We add a node corresponding to each EPT for gateways from which that fleet type may operate (e.g., node a). From each EPT node, we consider all possible movements to gateway locations at which that fleet type can land. The arrival time at the second location is the first gateway's EPT plus the aircraft's block time (i.e., flying time plus taxi time). We place a node in the time-space network for the location and its corresponding arrival time (e.g., node b) and connect the two nodes with a flight arc.

Next, we consider arrivals into hubs. We place a node in the time-space network corresponding to the sort start time at each hub (e.g., node d). We determine the time at which the plane must depart a gateway location in order to arrive at the hub by the sort start (i.e., sort start minus block time) and we add a node corresponding to that gateway and departure time to the time-space network (e.g., node c). We connect the two nodes with a flight arc.

When all flight arcs have been constructed for the pickup side of a given fleet type, we repeat the process for the delivery side, then repeat the process for all fleet types. For each location, ground arcs are added between successive nodes, such as arc (b, c) and all other horizontal arcs.

Finally, wrap-around arcs connect each location's LDT node on the delivery side to its EPT node on the pickup side (e.g., node a for gateway 2).

In the example shown in Figure 2-3, the network contains a path from gateway 1 to the hub. This path goes from gateway 1's EPT node through two ground arcs to a flight arc from gateway 1 to hub H. Similarly, there is a path from gateway 2 to the hub. No feasible route exists for 1-2-H as the first leg arrives at gateway 2 later than the departure of the second leg. Finally, the route from 2-1-H is feasible only if the duration between the two flight legs exceeds the minimum *turn time* for the given fleet type. This turn time is the duration required to prepare a fleet type for its next leg.

This *derived network* defines the how packages may flow from origin to destination and it provides the important link between the aircraft route variables and the package flow variables, as modeled in the forcing constraints. While aircraft routes have a limit on the number of flight legs, no such restrictions are placed on package flows. If the time available to transfer a package between planes exceeds a specified minimum transfer time, this transfer is allowed to happen. The result is a huge number of package flow variables. Reducing its size through node and link consolidation (see Hane et al. [42] and Kim et al. [52]) quickens the package flow variable generation process.

This method of construction "stretches" the routes to the extremes of their times windows. All feasible schedules for that route are represented by this "stretched" route. There are scheduling implications for this construct, as all flight arcs arriving to a hub do so at the same time. In practice this is not viable, so it is necessary to either force **ESSND** to create time-sequenced arrivals or to verify that solutions generated without this dynamic component have sufficient slack to manually create the appropriate time-sequencing of arrivals. This is explored in more detail in Chapter 4.

2.3.2 Carrier-Specific Origin-Destination Model

We revisit **ESSND** and take advantage of the route structure described above. We define the commodities by origin-destination pair and use the path-based decomposition strategy, as in **ESSND-OD**. We first present the notation that is used in the formulation:

Physical Assets

G	Set of gateway locations
H	Set of hub locations
F	Set of fleet types

Aircraft Route Notation

R_P	Set of pickup routes
R_D	Set of delivery routes
R	Set of routes ($R = R_P \cup R_D$)
R^f	Set of routes that can be flown by fleet type $f \in F$
$R(\overline{g})$	Routes originating at gateway $g \in G$
$R(\underline{g})$	Routes terminating at gateway $g \in G$

Package Flow Notation

K	Set of O-D commodities
P^k	Set of origin-destination paths available for commodity k
P	Set of all origin-destination paths, $P \equiv \bigcup_{k \in K} P^k$

Right Hand Side Data

b^k	Demand volume (in packages) of commodity k
n^f	Number of aircraft of type f
a_h	Number of planes that can land at hub $h \in H$
u_r^f	Capacity (in packages) of aircraft of type f flying route r

Indicators

$$\delta_{ij}^r = \begin{cases} 1 & \text{if route } r \in R \text{ contains arc } (i,j) \in A \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{ij}^p = \begin{cases} 1 & \text{if path } p \in P \text{ contains arc } (i,j) \in A \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{h}^p = \begin{cases} 1 & \text{if path } p \in P \text{ passes through hub } h \\ 0 & \text{otherwise} \end{cases}$$

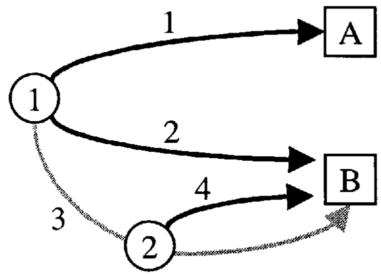

Figure 2-4: Routes for Route Set Notation Example

Decision Variables and Costs

y_r^f Number of planes of type f assigned to route r

x_p^k Fraction of packages of commodity $k \in K$ flown on path $p \in P^k$

d_r^f Cost of flying route r with fleet type $f \in F$

To clarify the notation used to represent the routes, we introduce the following example.

Example 2 (Route Set Notation Example) *The set operators may be combined. For instance, the set R_D^f is the set of delivery routes that can be flown by fleet type f, and $R_P^f(\overline{g})$ is the set of pickup routes departing from gateway g and flown by fleet type f. The simple network in Figure 2-4 shows four pickup routes. The gateways are labeled 1 and 2 and the hubs are labeled A and B. The route sets are the following: $R_P = \{1, 2, 3, 4\}$, $R_P(\overline{1}) = \{1, 2, 3\}$, $R_P(\overline{2}) = \{4\}$, $R_P(\underline{A}) = \{1\}$, and $R_P(\underline{B}) = \{2, 3, 4\}$.*

The carrier-specific express shipment service network design formulation, denoted by **ESSND-C**, is:

$$\min \sum_{f \in F} \sum_{r \in R^f} d_r^f y_r^f \tag{2.32}$$

subject to:

$$\sum_{k \in K} \sum_{p \in P^k} \delta_{ij}^p b^k x_p^k \leq \sum_{f \in F} \sum_{r \in R^f} \delta_{ij}^r u_r^f y_r^f \qquad \text{for all } (i,j) \in A \tag{2.33}$$

$$\sum_{p \in P^k} x_p^k = 1 \qquad \text{for all } k \in K \tag{2.34}$$

$$\sum_{k \in K} \sum_{p \in P^k} \delta_h^p b^k x_p^k \leq e_h \qquad h \in H \tag{2.35}$$

$$\sum_{r \in R_P^f(\overline{g})} y_r^f - \sum_{r \in R_D^f(\underline{g})} y_r^f = 0 \qquad \text{for all } g \in G,\ f \in F \tag{2.36}$$

$$\sum_{r \in R_D^f(\overline{h})} y_r^f - \sum_{r \in R_P^f(\underline{h})} y_r^f = 0 \qquad \text{for all } h \in H,\ f \in F \tag{2.37}$$

$$\sum_{r \in R_P^f} y_r^f \leq n_f \qquad f \in F \tag{2.38}$$

$$\sum_{r \in R_D^f} y_r \leq n_f \qquad f \in F \tag{2.39}$$

$$\sum_{f \in F} \sum_{r \in R_P^f(\underline{h})} y_r^f \leq a_h \qquad h \in H \tag{2.40}$$

$$x_p^k \geq 0 \qquad \text{for all } p \in P^k,\ k \in K \tag{2.41}$$

$$y_r^f \in \mathbb{Z}_+ \qquad \text{for all } r \in R^f,\ f \in F. \tag{2.42}$$

Constraint (2.33) forces the flow on each arc to be less than the capacity allocated on that arc. We ensure that all demand is flown through the convexity constraints (2.34). Each hub's sorting capacity, in number of packages, is ensured by constraint (2.35). Constraints (2.36) and (2.37) require that the number of planes departing a location is the same as the number arriving to that location, for gateways and hubs respectively. Constraints (2.38) and (2.39) limit the number of planes of each fleet type used on the pickup and delivery sides (with the balance constraints, one set is redundant). Each hub has a limit on the number of aircraft arriving during the sorting period (2.40). This constraint can be replaced with a dynamic version that enforces arrival limits within specified time intervals (see Chapter 4). The nonnegativity of package flow variables and the integrality of the aircraft route variables are enforced by constraints (2.41) and (2.42).

2.3.3 Solution Strategies

Testing **ESSND-C** on realistic problem instances (see Kim et al. [52], and Kim [50]) reveals that the LP relaxation provides a weak and ineffective bound for finding integer solutions via branch-and-bound. Strengthening the LP relaxation with valid inequalities results in an intractable model due to its size. To reduce the problem size, a related optimization model *consisting only of aircraft route variables* is solved. We describe these steps as follows.

Valid Inequalities

Valid inequalities are derived from aggregate capacity-demand inequalities. We partition the nodes of the derived network (N), such that $S \cup T = N$ and $S \cap T = \emptyset$, and we denote the arcs from S to T as the $[S,T]$ cut. Let $Y^f_{S,T}$ be the total number of aircraft of type f flying from set S to set T and let $D_{S,T}$ be the total demand originating in S and destined for T. Any feasible solution to the network design problem satisfies the following aggregate capacity-demand constraints:

$$\sum_{f \in F} u^f Y^f_{S,T} \geq D_{S,T} \quad \text{for any } [S,T] \text{ cut.} \tag{2.43}$$

Kim et al. [52] apply two types of cuts to strengthen these constraints. The first procedure is Chvátal-Gomory rounding (see Nemhauser and Wolsey [64]) and the second involves cutset inequalities designed for the two-facility network loading problem (see Magnanti, Mirchandani, and Vachani [58]). Kim et al. [52] show that neither type of cut dominates the other and, for the case of two fleet types, develops a rule to select the stronger inequality for a given $[S,T]$ cut. Due to the large number of inequalities, cutset inequalities and C-G cuts are found only for $|S| \leq 3$ or $|T| \leq 3$.

Optimization-Based Preprocessing

With the addition of cuts, the problem size grows and, for realistic problem instances, the root node LP cannot be solved within computing memory limits. To reduce the problem size, namely the number of design variables, Kim et al. [52] present a model consisting only of the aircraft route variables. This formulation is simply the portion of **ESSND** that involves the aircraft routing decisions. In addition, it is tightened by the addition of the aggregate capacity-demand constraints and their corresponding valid inequalities. In the absence of package flow variables, these capacity-demand constraints serve as a proxy for constraints (2.33) and (2.34). This model's solution is not guaranteed to be feasible with respect to **ESSND**. It does, however, guide the selection of the aircraft routes over which **ESSND** is optimized. Krishnan et al. [53] refine this selection process by solving an integer multicommodity flow problem that fixes a portion of the overall network solution, leaving a smaller problem to be solved by **ESSND**. In our experience, the tractability of the reduced **ESSND** is highly sensitive to the manner in which one makes this selection.

2.4 Summary

Initial work on the multi-hub, multi-fleet Express Shipment Service Network Design (**ESSND**) problem is based on traditional network design formulations in which aircraft route decisions and package flow decisions are modeled explicitly. Poor lower bounds and poor integer solutions lead to the use of general polyhedral methods (Chvátal-Gomory cuts) along with those specific to network design (cutset inequalities). The addition of these cuts and the resulting increase in problem size creates memory demands that can not be met with high-end workstations. By solving an integer program containing only aircraft route variables and their associated constraints, the set of variables over which the network design model is solved is reduced and is tractable for certain **ESSND** problem instances.

Testing these solution approaches yielded several critical observations. First, the **ESSND** approach has difficulty with high congestion levels and it can not generate an integer solution on recent test problem instances. Second, the running time for the aircraft-only (preprocessing) model is considerably faster than **ESSND**, though the aircraft-only model is not guaranteed to generate a feasible solution to the **ESSND** problem. Third, experimentation using *set covering constraints* to enforce carrier-specific connectivity requirements tended to increase the integrality of the LP relaxation. These observations are important in the development of a new formulation that has the benefits of optimizing only over design variables and avoids the weaknesses of traditional network design formulations and solution approaches.

Chapter 3

Composite Variable Formulation for the Express Shipment Service Network Design Problem

The Express Shipment Service Network Design (**ESSND**) problem, as presented in the previous chapter, poses computational challenges that can not be overcome on problems of the size we need to consider. At the root of these problems are the tendency for the solutions of the LP relaxation to fly fractional aircraft and the presence of aircraft conservation of flow constraints. Any localized use of fractional aircraft spreads through the network as a consequence of these constraints, exacerbating the fractionality of the LP relaxation and weakening its bound on the optimal integer solution.

Practical solution methods must be robust in the sense that obtaining a solution should depend neither on the data nor on the person using the model. Running time of the model is also an important consideration, especially if such a model is to serve as the core element of broader planning problems, such as a combined Next-Day Air (NDA) and Second-Day Air (SDA) system or the combined domestic and international networks.

With this in mind, we present a new formulation strategy for solving network design problems that uses only *design variables.* We refer to the general strategy as a *composite variable formulation* and the specific formulation for the express shipment problem as the Aircraft Routing Model (**ARM**). We demonstrate how feasible aircraft routes are constructed and how they are combined into *composite* decision variables. Finally, we highlight how this formulation strategy allows us to consider operating constraints that are difficult to capture in traditional network design formulations such as **ESSND**.

3.1 Planning Framework

Models and algorithms to solve the **ESSND** problem will be at the core of a larger decision support framework at the United Parcel Service (UPS), as depicted in Figure 3-1. The overall objective is to give network planners a useful design tool that pulls together data, feeds it to the model, and returns the model's solution in a meaningful form. The first component, volume compression, takes the customer-to-customer demands and determines origin-destination demands for all gateways within the Next-Day Air network. Using hub service territories determined manually, volume to and from each gateway is assigned for processing at one or more hub. This provides *gateway-hub pair demands* (versus gateway-gateway demands) that are the commodities input to **ARM**. The model outputs the routes, fleetings, and the feasible package flows. From the model's solution, a schedule is created, package flows are determined (for the customer-to-customer demands), and aircraft-specific tail numbers are assigned to the routes.

The context of this system is to support the planners who develop annual routing plans for the carrier and the planners who modify the current year plan to respond to major changes in the system (e.g., the addition or deletion of major accounts). Additional uses include supporting strategic analyses of new aircraft purchases, increases in hub sort capacity, increases in airport landing capacity, and additions and closures of hub or gateway locations. A network design model with sufficiently small run times might also serve as a core component of a daily re-planning and operations recovery system.

The current planning process is largely manual. While planners use computers to manage large amounts of data, they rely on their experience as the primary tool for creating aircraft routing plans. The long-range planners typically get only a couple of passes through the data in the six months it takes to iteratively build a plan. A decision support system would enable the planner to build a network design with lower operating cost in less time and to iterate

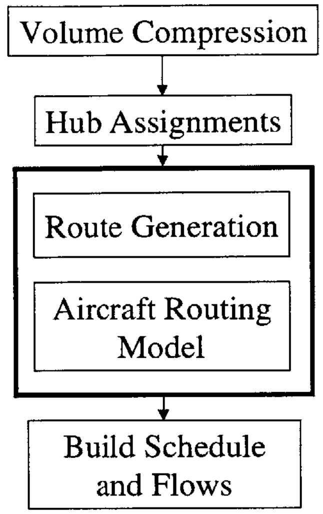

Figure 3-1: Air network planning architecture

through the planning process many times during the same six month period. This would also enable the planner to perform scenario analyses to compensate for any uncertainty in demand forecasts.

3.2 Composites, Covers, and Reformulation

The fundamental idea behind the new formulation strategy is to formulate the **ESSND** problem in terms of the design variables and to implicitly capture the package flows within the new variables. We construct our variables by forming *composites* that comprise one or more aircraft routes in a single decision variable. The composites that we include are those whose capacity exceeds the demand for gateway-hub pairs incident to aircraft routes contained in the composite variable. We motivate the idea of composites through the following example:

Example 3 *Consider the simple network shown in Figure 3-2. Our objective is to move 6000 packages from g to h using some combination of two fleet types. The low-capacity aircraft has a capacity of 5000 packages and the high-capacity aircraft has a capacity of 7000 packages.*

Typical network design models have variables for package flows and variables for aircraft route (i.e., design) decisions. Instead, we recognize that, in this example, we do not need to

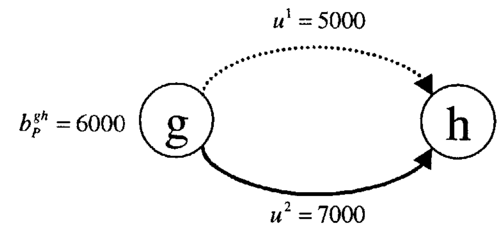

Figure 3-2: Simple two-node network with two aircraft routes

model the package flows explicitly. Rather we can create a model in terms of the capacity that we make available to **cover** *the demand.*

To ensure adequate capacity we write the **capacity-demand** *constraint:*

$$5000y_1 + 7000y_2 \geq 6000.$$

Without requiring integral values for $\mathbf{y}$, *minimizing cost (which we have not specified) would force a fractional solution, either* $y_1 = \frac{6}{5}$ *or* $y_2 = \frac{6}{7}$.

Next, we add a constraint based on the intuitive idea that we require at least one plane to fly from g to h. That is, we have the following **covering** *constraint:*

$$y_1 + y_2 \geq 1.$$

This constraint removes some fractionality from the original constraint ($y_2 = \frac{6}{7}$ *is no longer feasible, for example). However, this constraint alone does not ensure a feasible solution (e.g.,* $y_1 = 1$ *does not allow a flow of 6000 packages).*

Consider the case when the capacities are larger than the demand. For example, let $u^1 =$ 8000. *Then the capacity-demand constraint is of the form:*

$$8000y_1 + 7000y_2 \geq 6000.$$

We can reduce the capacity coefficients to 6000 and maintain the feasibility of all integer solutions. The resulting constraint is equivalent to the covering constraint. We remove the original capacity-demand constraints because it is dominated by the covering constraint.

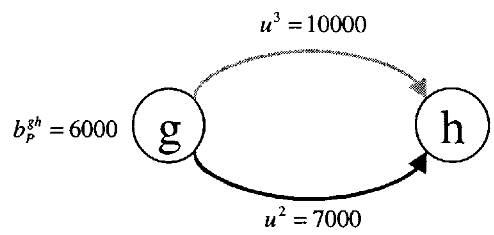

Figure 3-3: Simple two-node network with composite variables

A model stated purely in terms of covering constraints has computational benefits. As we have currently modeled the problem, we only take advantage of this when all capacities exceed demands. We are able to drive the model to this form by considering combinations of aircraft routes that are grouped together and treated by the model as a single object. We call these objects ***composites****.*

In the network shown in Figure 3-2, we see that selecting one type 2 aircraft covers the demand. However, covering the demand with fleet type 1 requires ***two*** *aircraft. We create a new variable that represents two type 1 aircraft with an effective capacity of 10000 packages (see Figure 3-3). The capacity demand constraint becomes*

$$7000y_2 + 10000y_3 \geq 6000,$$

which may then be transformed to a covering constraint through coefficient reduction. Because the covering constraint dominates the capacity-demand constraint, we remove the capacity-demand constraint from the model and the fractionality is removed.

This example illustrates two key observations that lead to the composite variable formulation. First, we model the capacity-demand relationship using capacity-demand constraints, allowing us to remove the explicit representation of the package flow variables. Second, we tighten the formulation by combining aircraft routes so that capacity-demand constraints can be reduced to covering constraints. Keeping these ideas in mind, we next develop a composite variable formulation for the full **ESSND** problem.

3.2.1 Aircraft Routing Model (ARM) Formulation

The model we describe is initially catered to the carrier-specific problem we described in the previous chapter. Two assumptions that we continue to enforce are that package flow costs are zero and commodities are specified by gateway-hub pairs for both the pickup side and the delivery side. We begin by introducing the relevant notation consistent with notation defined in Chapter 2:

G	Set of gateway locations
H	Set of hub locations
F	Set of fleet types
R	Set of routes
R^f	Set of routes flown by fleet type $f \in F$

Relevant data includes:

n_f	Number of aircraft of type $f \in F$
a_h	Number of planes that may land at hub $h \in H$
u_r^f	Capacity (in packages) of aircraft of type $f \in F$ flying route $r \in R^f$
b_P^{gh}	Pickup demand (in packages) from gateway $g \in G$ to hub $h \in H$
b_D^{gh}	Delivery demand (in packages) from hub $h \in H$ to gateway $g \in G$
δ_{ij}^{fr}	=1 if aircraft route (f, r) contains arc (i, j) in the derived time-space network.

Definition 4 *A **composite**, c, is a set of distinct aircraft routes (f, r), $f \in F$, $r \in R^f$. Associated with c are the parameters γ_c^{fr}, which indicate the number of planes of fleet type f that fly route r in composite c.*

To store which routes, gateways, and hubs are included in a composite, we introduce the following sets. Let A_c be the set of arcs contained in composite c. Let P_c^{gh} denote the set of paths from g to h using arcs in A_c. Define x_p^{gh} as the fraction of the commodity (g, h) demand flown on path $p \in P_c^{gh}$. Using b^{gh} to represent demand on either the pickup side or the delivery side, we introduce the following definition of *composite covers.*

Definition 5 *Let $\mathcal{K}_c$ be a set of gateway-hub commodities (either on the pickup side or the delivery side), each commodity having demand b^{gh}. A composite, c, is said to* **cover** *$\mathcal{K}_c$ if a feasible flow exists for all gateway-hub demands contained in $\mathcal{K}_c$. That is:*

$$\sum_{(g,h)\in\mathcal{K}_c}\sum_{p\in P_c^{gh}} \delta_{ij}^{p} b^{gh} x_p^{gh} \leq \sum_{(f,r)\in c} \delta_{ij}^{fr} u_r^f \gamma_c^{fr} \quad (i,j)\in A_c \tag{3.1}$$

$$\sum_{p\in P_c^{gh}} x_p^{gh} = 1 \quad (g,h)\in\mathcal{K}_c. \tag{3.2}$$

The interpretation of a composite cover is that it is a collection of routes that has capacity to carry all demands between some specified set of gateways and hubs (either on the pickup side or delivery side). Note the similarity of these conditions to the forcing and convexity constraints of the network design formulation (**ESSND-OD**) presented in Chapter 2.

The definition of composite covers leaves the possibility that one or more aircraft routes in the composite are not needed to cover the demands. In other words, those aircraft are flying empty. To avoid an explosion in the number of unnecessary composite variables, we introduce the notion of a *minimal composite.*

Definition 6 *The composite c is said to be a* ***minimal cover of*** *$\mathcal{K}_c$ if, by removing any aircraft route from c, it no longer covers the set of demands in $\mathcal{K}_c$.*

The following example illustrates these three definitions.

Example 7 *Consider the two node network shown in Figure 3-4. There is a single route (1) that can be flown by two different fleet types. The family of minimal composite covers is given by $\{\{(2,1)\},\ \{(1,1)\}\}$, with $\gamma_1^{2,1} = 1$ and $\gamma_2^{1,1} = 2$. In other words, the first composite cover is defined by a single aircraft of type 2 flying from g to h. The second composite cover contains two aircraft of fleet type 1 flying from g to h and providing adequate capacity to cover the demand. The set $\mathcal{K}_c$ is given by $\{(g,h)\}$. A third composite cover is $\{(1,1),(2,1)\}$ with $\gamma_3^{1,1} = 1$ and $\gamma_3^{2,1} = 1$. It is not minimal since the removal of aircraft route $(1,1)$ yields a cover of $\mathcal{K}_c$.*

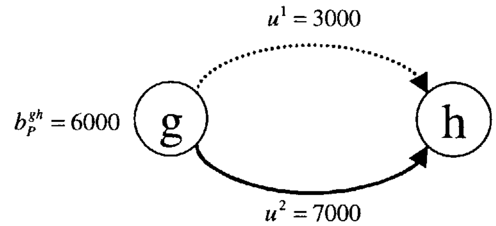

Figure 3-4: Example of a simple composite

We recast the carrier-specific network design formulation (**ESSND-C**) using only decision variables corresponding to composites. We define the following sets:

$\mathcal{C}$	Family of composite covers
$\mathcal{C}_P$	Family of composite covers for the pickup side
$\mathcal{C}_D$	Family of composite covers for the delivery side.

We define the decision variables to be:

v_c	Binary variable for including composite c

and the indicators:

γ_c^f	Number of fleet type f included in composite c
$\gamma_c^f(\overline{g})$	Number of fleet type f departing gateway g included in composite c (similarly for hub h)
$\gamma_c^f(\underline{g})$	Number of fleet type f arriving to gateway g included in composite c (similarly for hub h)

$$\delta_c^{gh} = \begin{cases} 1 & \text{if } b^{gh} \text{ is covered by } c \text{ (i.e., } (g,h) \in \mathcal{K}_c) \\ 0 & \text{otherwise.} \end{cases}$$

The cost of a composite is the sum the costs of aircraft routes contained in the composite:

$$d_c = \sum_{(f,r)\in c} \gamma_c^{fr} d_r^f.$$

We seek to minimize the cost of composites (hence, aircraft routes) subject to aircraft routing restrictions and the need to satisfy all pre-assigned gateway-hub demands. This leads to the following optimization model (**ARM**), which we show in Chapter 5 to be equivalent to the original formulation under the presence of the gateway-hub demand assignments and no package flow costs:

$$\min \sum_{c \in \mathcal{C}} d_c v_c \tag{3.3}$$

subject to

$$\sum_{c \in \mathcal{C}_P} \gamma_c^f(\overline{g}) v_c - \sum_{c \in \mathcal{C}_D} \gamma_c^f(\underline{g}) v_c = 0 \qquad \text{for all } g \in G,\ f \in F \tag{3.4}$$

$$\sum_{c \in \mathcal{C}_P} \gamma_c^f(\underline{h}) v_c - \sum_{c \in \mathcal{C}_D} \gamma_c^f(\overline{h}) v_c = 0 \qquad \text{for all } h \in H,\ f \in F \tag{3.5}$$

$$\sum_{c \in \mathcal{C}_P} \gamma_c^f v_c \le n_f \qquad f \in F \tag{3.6}$$

$$\sum_{c \in \mathcal{C}_D} \gamma_c^f v_c \le n_f \qquad f \in F \tag{3.7}$$

$$\sum_{f \in F} \sum_{c \in \mathcal{C}_P} \gamma_c^f(\underline{h}) v_c \le a_h \quad h \in H \tag{3.8}$$

$$\sum_{c \in \mathcal{C}_P} \delta_c^{gh} v_c = 1 \quad (g,h) : b_P^{gh} > 0 \tag{3.9}$$

$$\sum_{c \in \mathcal{C}_D} \delta_c^{gh} v_c = 1 \quad (g,h) : b_D^{gh} > 0 \tag{3.10}$$

$$v_c \in \{0,1\} \ \text{ for all } c \in \mathcal{C}. \tag{3.11}$$

The interpretation of this model is straightforward. We select composites such that the aircraft usage satisfies aircraft constraints (3.4) through (3.8), which are identical to the aircraft constraints in the network design formulations introduced in Chapter 2. Furthermore, composites must be selected so that all nonzero gateway-hub demands are covered for both pickup (3.9) and delivery (3.10). Each composite contained in a set partitioning constraint is guaranteed to cover the demand corresponding to that constraint and selecting a set of composites that satisfies all constraints will cover the demand for all gateway-hub pairs.

Practical considerations dictate several changes to this formulation. The following sections

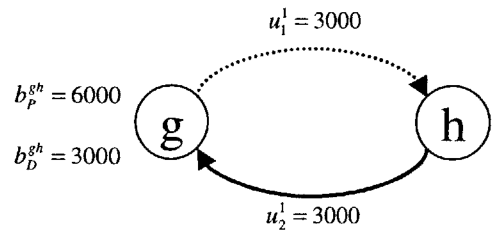

Figure 3-5: Single-gateway, single-hub, single-fleet example with demand imbalance

address the issues of ferrying, aircraft balance, and general aircraft route capacities. All such changes are required for **ARM** to be an appropriate model for realistic instances of this problem.

3.2.2 Modeling Empty Aircraft Movements

Generating minimal composites has an impact on the model's ability to *balance* aircraft. Demands are often not symmetric across the pickup and delivery sides. In order to maintain balance in such cases, it may be necessary to fly empty aircraft between gateways. These repositioning moves are known as a *ferry routes.*

Minimal composites, by definition, do not include ferry routes. We can model ferries by either allowing a composite variable to be selected more than once or by introducing new variables to represent empty plane movements. Without these changes, set partitioning constraints (3.9) and (3.10) may yield an infeasible problem, as illustrated in the following example.

Example 8 *Consider the single-fleet, single-gateway, single-hub example in Figure 3-5. We consider the case when using minimal composites. On the pickup side, the set of minimal composites is* $C_P = \{\{(1,1)\}\}$*, with* $\gamma_1^{1,1} = 2$*. On the delivery side, the set of minimal composites contains a single composite that has a single aircraft route. That is* $C_D = \{\{(1,2)\}\}$*, with* $\gamma_2^{1,2} = 1$*. The balance constraint for fleet type 1 at gateway g is* $2v_1 - v_2 = 0$ *and the set partitioning constraints are* $v_1 = 1$ *and* $v_2 = 1$*. No simultaneous solution exists for these constraints. Replacing the set partitioning constraints with set covering constraints of the form* $v_1 \geq 1$*,* $v_2 \geq 1$*, and* $v_1,\ v_2 \in \mathbb{Z}^+$*, there exists a feasible solution, namely* $v_1 = 2$ *and* $v_2 = 1$*.*

To allow multiple selections of composites, we modify the formulation with covering (versus set partitioning) constraints and general integer (versus binary) variables. Constraints (3.9) and (3.10) become

$$\sum_{c \in \mathcal{C}_P} \delta_c^{gh} v_c \geq 1 \quad (g,h) : b_P^{gh} > 0 \tag{3.12}$$

$$\sum_{c \in \mathcal{C}_D} \delta_c^{gh} v_c \geq 1 \quad (g,h) : b_D^{gh} > 0 \tag{3.13}$$

and the restriction on the decision variables becomes

$$v_c \in \mathbb{Z}_+ \text{ for all } c \in \mathcal{C}. \tag{3.14}$$

The downside of this change is that if a composite is selected more than once, its repeat selection implies that *all* aircraft routes in the composite are flown empty. However, it may have been more advantageous to ferry a single aircraft route in the composite. To allow this type of repositioning, we incorporate new decision variables corresponding to ferry routes. We provide two options. First, we include single-leg aircraft routes between gateways and hubs that are *not* included in the original composite list. Second, we include single-leg gateway-to-gateway ferry routes, which are also *not* included in the original composite list since they neither originate nor terminate at a hub. We model ferry routes with the following decision variables:

ϕ_{ij}^f Number of aircraft f ferried from location i to location j.

The balance constraints (3.4) and (3.5) are replaced by:

$$\sum_{c \in \mathcal{C}_p} \gamma_c^f(\overline{g}) v_c + \sum_{i \in G \cup H} \phi_{gi}^f - \sum_{c \in \mathcal{C}_D} \gamma_c^f(\underline{g}) v_c - \sum_{i \in G \cup H} \phi_{ig}^f = 0 \quad \forall\, g \in G,\ f \in F \tag{3.15}$$

$$\sum_{c \in \mathcal{C}_p} \gamma_c^f(\underline{h}) v_c + \sum_{g \in G} \phi_{gh}^f - \sum_{c \in \mathcal{C}_D} \gamma_c^f(\overline{h}) v_c - \sum_{g \in G} \phi_{hg}^f = 0 \quad \forall\, h \in H,\ f \in F. \tag{3.16}$$

Note that these ferry routes are, in fact, composites, because each such route covers the demand of the empty gateway-hub set. Their cost is determined like a regular route, as the sum of the cycle cost and the operating cost for flying fleet type f between the two locations.

3.2.3 Balance Constraint Modifications

The planners' solutions balance almost everywhere. The imbalances in the Next-Day Air (NDA) network are corrected during the Second-Day Air (SDA) network operations, ensuring that the NDA network resets itself for the next night of operations. We present modeling changes that allow imbalances at specified gateways (we call this *quasi-balance*). We also describe changes that allow solutions to be balanced against an existing SDA network. This forces the NDA solution to have some of its aircraft assigned to given locations at the beginning and end of the NDA network.

Quasi-Balance at Gateways

To take advantage of the proximity of two or more gateways, it may be desirable to balance departures from one gateway using arrivals to another gateway. In fact, this situation may be forced by operational restrictions such as noise curfews at airports. Planes would then be repositioned during the SDA operation. For example, in the Los Angeles basin, all pickup routes departing from a set of five gateways must be offset by arriving delivery routes to one (or more) gateway in the region. We force all landings in that *neighborhood* to occur at one gateway by setting the delivery demand into that one gateway equal to the neighborhood's total delivery demand. All other gateways in the neighborhood would have zero delivery demand.

To model quasi-balance, we designate a single gateway, g, to represent the entire neighborhood. Let $Q(g)$ be the set of gateways in that neighborhood, including g. For any gateway contained in another gateway's neighborhood, $Q(g) = \emptyset$. We update the gateway balance constraints (3.15) as follows:

$$\sum_{j \in Q(g)} \left[\sum_{c \in \mathcal{C}_p} \gamma_c^f(\bar{j}) v_c + \sum_{i \in G \cup H} \phi_{ji}^f - \sum_{c \in \mathcal{C}_D} \gamma_c^f(\underline{j}) v_c - \sum_{i \in G \cup H} \phi_{ij}^f \right] = 0 \quad \forall\, g \in G,\ f \in F.$$

For gateways not included in any neighborhood, we define $Q(g) = g$, and summing over the elements of $Q(g)$ yields the original gateway balance constraints (3.15).

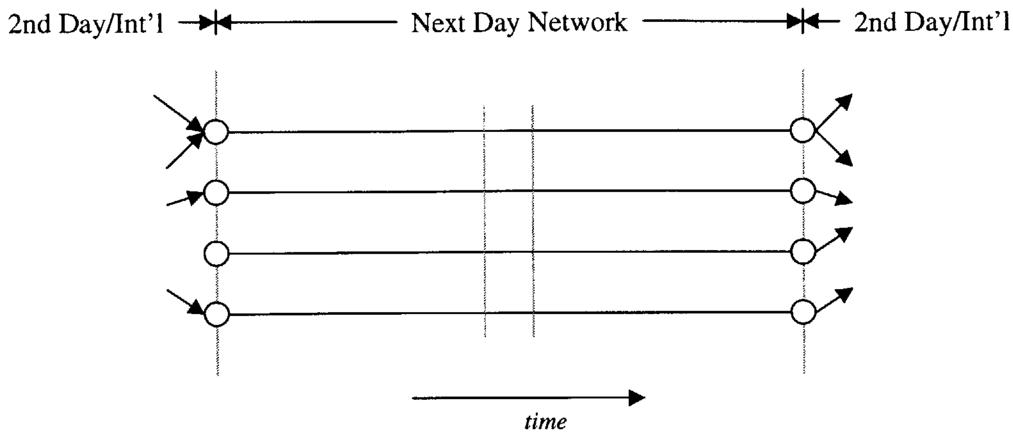

Figure 3-6: Boundary conditions imposed by the Second Day Air (SDA) network

Balancing Against the Second Day Air Network

A solution that ensures balance at the beginning and end of the overnight network ignores that fact that aircraft are used during the day for domestic Second-Day Air (SDA) and international operations (see Figure 3-6). The positioning of aircraft in the NDA network solution must be consistent with the positioning of aircraft in the SDA and international networks. Hence, we can treat the SDA network as fixed and anchor the start and end of the NDA network with aircraft requirements at each gateway location.

To account for these boundary conditions, we take the following as input:

- α^f_{gP} Number of aircraft of fleet type $f \in F$ located at gateway $g \in G$ at the beginning of the Next-Day Air network (i.e., the end of the SDA network)
- α^f_{gD} Number of aircraft of fleet type $f \in F$ located at gateway $g \in G$ at the end of the Next-Day Air network (i.e., the beginning of the SDA network)

Balancing against these boundary conditions, **ARM** must provide the option of keeping the aircraft on the ground for the duration of the NDA network. Any such grounded aircraft must be included in the plane count constraints. Next, **ARM** must not limit the next-day air network to use only the aircraft that are used in the SDA plan (the SDA network uses only a fraction of the aircraft needed in the NDA network). For these additional aircraft, **ARM** must ensure gateway balance. Finally, we must still allow ferry routes to be used to reposition aircraft in the next day network.

We introduce ground arcs for each gateway location and fleet type. The upper bound on ground arc flow is $\alpha^f_{g_P}$. We also introduce wrap-around arcs that ensure aircraft not flown in the SDA network will be properly repositioned prior to the start of the Next-Day network.

v^f_g Number of aircraft of type f that remain on ground at location g during the Next Day network (ground arc)

ω^f_g Number of aircraft of type f that remain on ground at location g during the 2nd Day network (wrap-around arc)

We split the ferry variables into pickup side and delivery side ferry variables

$\phi^f_{ij_P}$ Number of aircraft f ferried from gateway i to gateway j prior to a pickup route

$\phi^f_{ij_D}$ Number of aircraft f ferried from gateway i to gateway j following a delivery route

We replace the set of aircraft gateway balance constraints (3.4) with two sets of constraints:

$$\sum_{c\in\mathcal{C}_P} \gamma^f_c(\overline{g})v_c + \sum_{i\in G}\phi^f_{gi_P} - \sum_{i\in G}\phi^f_{ig_P} + v^f_g - \omega^f_g = \alpha^f_{g_P} \quad g\in G,\ f\in F \tag{3.17}$$

$$\sum_{c\in\mathcal{C}_D} \gamma^f_c(\underline{g})v_c + \sum_{i\in G}\phi^f_{ig_D} - \sum_{i\in G}\phi^f_{gi_D} - v^f_g + \omega^f_g = \alpha^f_{g_D} \quad g\in G,\ f\in F. \tag{3.18}$$

On the pickup side, for each fleet type and each gateway, (3.17) ensures that the number of planes departing the gateway on pickup or ferry routes plus the number of aircraft grounded at the gateway during NDA operations must equal the number of planes at that gateway at the end of the SDA network plus the number of planes grounded at the gateway from the previous day's NDA network plus the number of planes that ferry to the gateway prior to the start of pickup routes. On the delivery side, (3.18) ensures that for each fleet type and each gateway, the number of planes arriving at the location (on delivery routes, ground arcs, and ferry routes) must equal the number of planes leaving the gateway on ferry routes or SDA routes plus those planes that remain at the gateway until the start of the next day's NDA operations.

With the possibility of grounding aircraft during the NDA network, we modify the plane count constraints (3.6) and (3.7) to include the ground arcs:

$$\sum_{c \in \mathcal{C}_P} \gamma_c^f v_c + \sum_{g \in G} v_g^f \leq n^f \qquad f \in F \tag{3.19}$$

$$\sum_{c \in \mathcal{C}_D} \gamma_c^f v_c + \sum_{g \in G} v_g^f \leq n^f \qquad f \in F. \tag{3.20}$$

The hub balance constraints (3.5) force an equal number of planes to be used on the pickup side and the delivery side. Therefore, either constraint set (3.19) or (3.20) can be removed because they are redundant.

3.2.4 General Aircraft Capacities

Currently, aircraft capacities are constant across all routes except west-coast pickup routes. Capacities of planes flying pickup routes from west coast locations to non-west coast hubs are reduced by 30 percent. This reduction is a rule-of-thumb used by planners to account for the tighter time windows involved with handling west coast volume. The capacity reduction effectively doubles the number of fleet types and, therefore, doubles the number of C-G cuts applied to **ESSND-C** (see Chapter 2). The cutset inequalities (see Magnanti et al. [58]) do not generalize because capacities are not integer multiples of each other. Recent results due to Atamtürk [4] have characterized valid inequalities for the case of general capacities, but they have not been used in **ESSND-C**. **ARM**, however, is *not* affected as the model's structure does not rely on specific values for u_r^f.

Future modeling requirements dictate that capacity be a function of distance. For each fleet type, a range-payload curve, as shown in Figure 3-7, specifies the capacity (in packages) as a function of block hours. A plane flying a longer route must carry more fuel, reducing the number of package it can carry. If, instead of packages, the demand volumes are measured in containers, this curve is a staircase (rather than piecewise linear) function. Regardless of its continuity, the function is a major obstacle in applying polyhedral results to network design formulations. C-G cuts become computationally burdensome as the number of capacity-distinct facilities may approach the total number of aircraft routes. And, for reasons described above, the cutset inequalities no longer apply. **ARM** easily handles the new capacities as they are

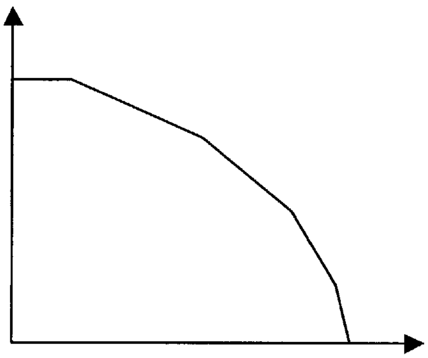

Figure 3-7: Continuous piecewise linear range-payload curve

simply input data and do not affect the composite generation procedures.

3.3 Generating Variables

The **ARM** formulation relies on our ability to characterize the complete family of composites. In practice, generating all composites might not be possible. By making initial assumptions on which types of composites are likely in the solution and by generating corresponding minimal composites, the resulting model's size is reasonable. With strong lower bounds provided by its LP relaxation, **ARM** quickly generates integer solutions without having to incorporate either column or row generation.

One consequence of this approach is that optimizing over this reduced set of composites might not yield the optimal solution to the original problem. An optimization-based approach is still attractive as the model provides a measure of optimality *given* the input set of composite variables. We can adjust how broadly we define the reduced set of composites based on our desired degree of optimality. In Chapter 5, we provide the theoretical basis for constructing the reduced set of composites as described in this chapter.

This formulation strategy allows us to handle integer (versus continuous) package flows, which is critical since network planners often work with containers rather than packages. Depending on its type, an aircraft may hold from 9 to 32 containers. Thus, the underlying multicommodity component of the network design problem should be modeled as *integral.* While **ARM** easily handles this requirement, the tractability of the network design formulation (**ESSND-C**) becomes a concern.

In this section, we describe the methods used for constructing this reduced set of composites, which we denote by $\mathcal{C}^r$. We present procedures for constructing and classifying feasible aircraft routes and how, from those routes, we build the composite variables over which **ARM** is optimized.

3.3.1 Aircraft Route Generation

Prior to building the composite variables, we must construct the fundamental building blocks: the aircraft routes. In the network design formulations of Chapter 2, routes were built from a time-space network, which provided the structure on which packages flows were connected to aircraft route decisions (through the forcing constraints). With the elimination of forcing constraints and package flow variables, the time-space network is no longer needed. Instead, we cycle through fleet types and gateway locations to find *feasible aircraft routes* (not including ferry routes), which we now define:

Definition 9 *A* ***route*** *is identified by a set of locations and is designated as pickup or delivery. The set of locations may contain two or three locations, at least one of which is a hub. A pickup route always terminates at a hub and a delivery route always originates at a hub.*

Definition 10 *An* ***aircraft route*** *is a route flown by a specific fleet type.*

Definition 11 *A* ***feasible*** *aircraft route is one that satisfies level-of-service requirements, fleet restrictions at locations, fleet-specific limits on number of legs per route, fleet-specific turn time requirements, and fleet-specific flying ranges.*

Feasible routes are constructed so that the first leg occurs as early as possible and the second leg as late as possible. This maximizes the time windows during which packages can be transferred between planes at intermediate stops on their routes.

For gateway i, we denote the *Early Pickup Time* for gateway i as $EPT(i)$ and the *Late Delivery Time* as $LDT(i)$. Travel time for fleet f between two locations i and j is denoted by $BlockTime(i, j, f)$. An aircraft's flying range is given by $Range(f)$ and the time it takes to turn the aircraft at intermediate locations is given by $TurnTime(f)$. The timing requirements for the hubs are specified by $SortStart(i, j, h)$. Sort times depend upon whether i and j are west coast locations and h is not a west cost hub, in which case the sort start time is later than for east coast arrivals into the same hub. On the delivery side, $SortEnd(i, j, h)$ depends upon the same criterion, as aircraft routes destined for west coast locations may depart earlier than those departing for east coast locations.

To generate all feasible routes, we loop through all combinations of fleet types ($f \in F$), hubs ($h \in H$), and gateway location pairs $\{(i, j) : i,\ j \in G\}$ in the hub's service territory. The 3-tuple (i, j, h), along with the designation of pickup or delivery, denotes a *route*. When $i = j$, the route contains just one leg. Combined with the fleet type, these data fully describe an *aircraft route*. The *feasibility* of the route is determined by the subroutines in Figures 3-8 and 3-9 for pickup routes and delivery routes, respectively.

As a result of this construction, all pickup routes "arrive" at the hub at the sort start time. While the model selects routes, fleetings, and package flows, it does not immediately output a viable schedule. The solutions have sufficient slack to allow proper staggering of planes arriving and departing the hub and gateway locations. An example of this is presented along with the computational results in Chapter 4. Alternatively, dynamic arrival constraints can be added to **ARM** to create this staggering.

3.3.2 Composite Variable Construction

The feasible aircraft routes, $\{(f, r) : r \in R^f,\ f \in F\}$, serve as the building blocks for composite variables. If an aircraft route, consisting of one or two gateways and a single hub, covers the demand of its gateway-hub pairs, it is, by definition, a *composite cover*. For aircraft routes that are not composite covers, we determine *a priori* the capacity they provide to a given gateway-hub pair and, for that pair, build a composite cover using pieces of excess capacity from multiple aircraft routes. We use similar methods to utilize the excess capacity on aircraft routes to transfer packages between aircraft routes at intermediate gateways.

The methods that we describe in this section exploit problem-specific features. A key observation is that most gateway-hub pairs can be satisfied with a single aircraft route incident

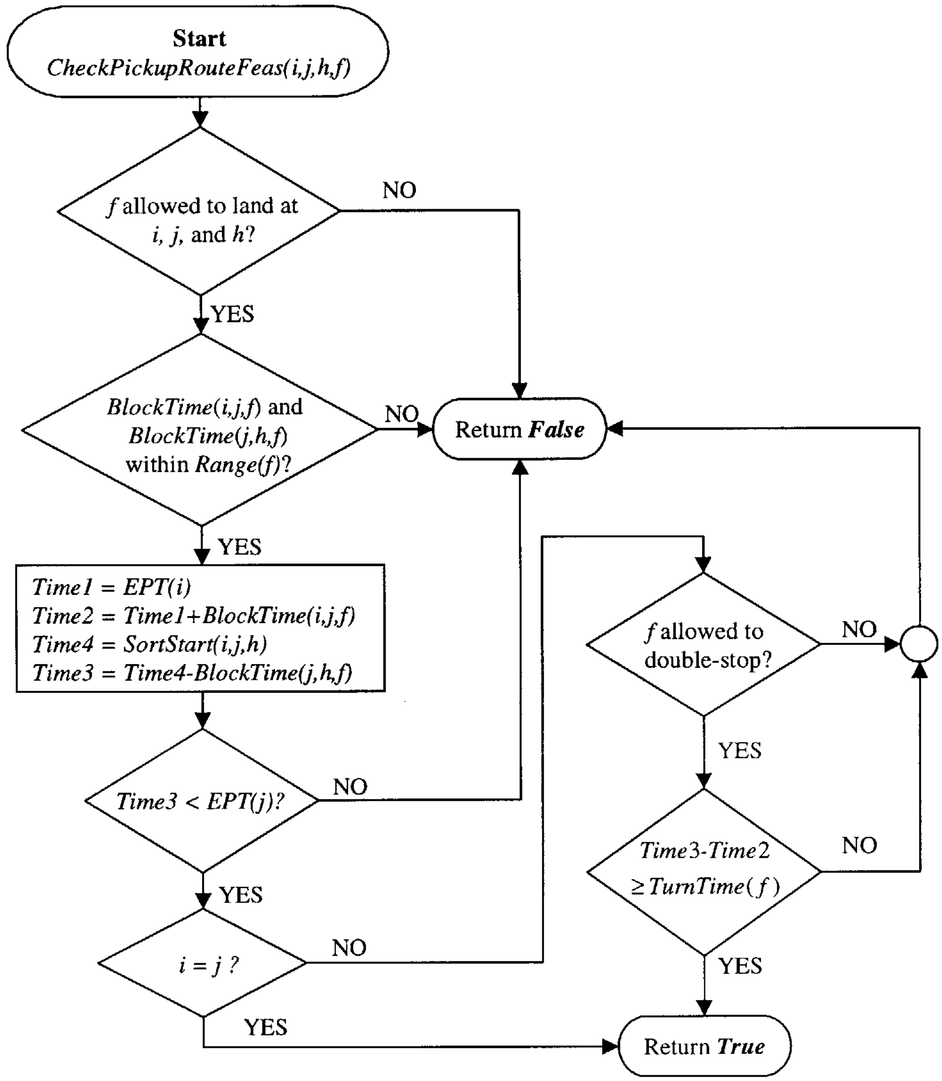

Figure 3-8: Pickup route feasibility check for gateways i and j, hub h, and fleet type f

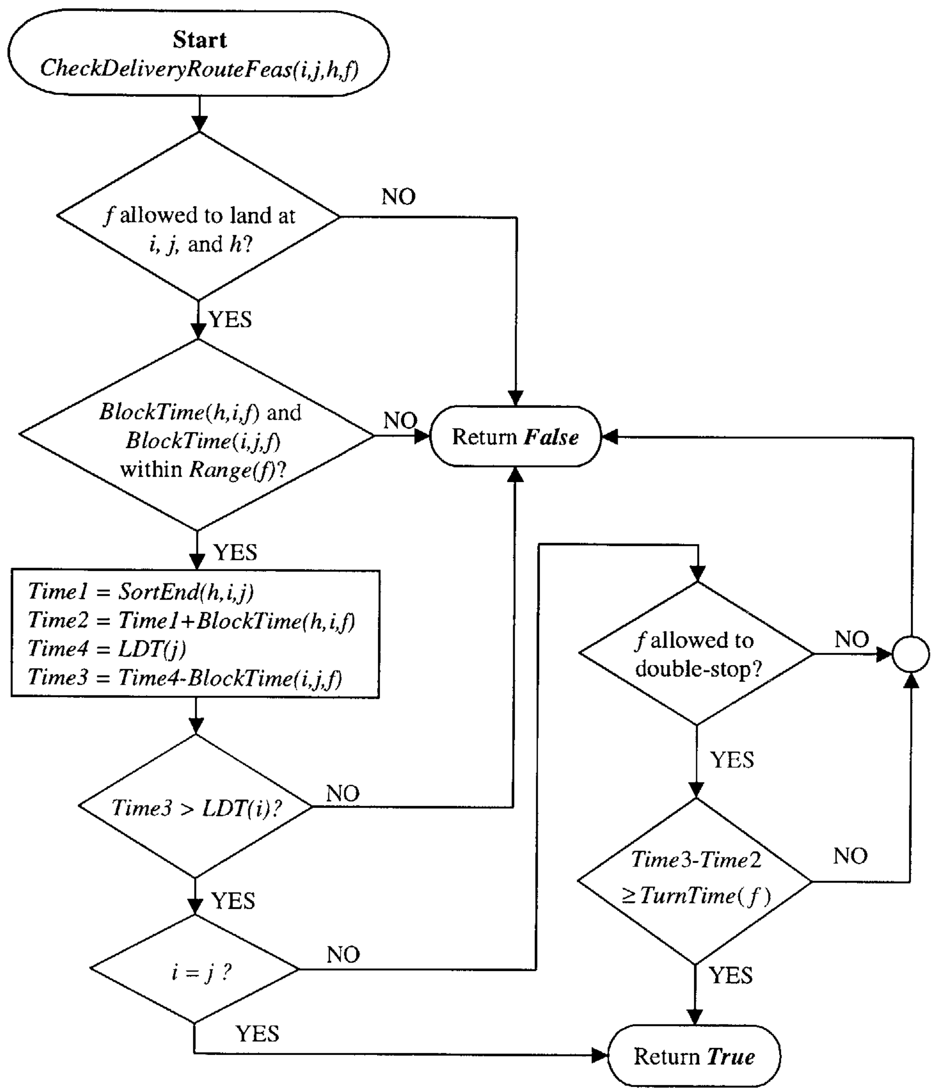

Figure 3-9: Delivery route feasiblity check for gateways i and j, hub h, and fleet type f

to the gateway. So rather than generating the family of all possible composites, we limit which and how many aircraft routes may be used to build composites. This leads to the development of simple, recursive subroutines that generate the composites, $\mathcal{C}^r$, that are then fed to **ARM**. Furthermore, to handle additional operational constraints, such as limits on ramp transfer volume, we may build simple rules into these composite-generating subroutines.

Composite generation focuses on each gateway-hub pair with nonzero demand. We define the following sets by gateway-hub pair:

$C(g,h)$ Composites that cover the demand for gateway-hub pair (g,h); the full set of composites is $\mathcal{C}^r \equiv \bigcup_{(g,h)\in\mathcal{K}} \mathcal{C}(g,h)$

$S(g,h)$ Aircraft routes guaranteed to cover (g,h) demand; the full set of *single route composites* is $S \equiv \bigcup_{(g,h)\in\mathcal{K}} S(g,h)$

$U(g,h)$ Aircraft routes providing partial capacity for (g,h) demand; the full set of partial capacity aircraft routes is $U \equiv \bigcup_{(g,h)\in\mathcal{K}} U(g,h)$

These sets are constructed as shown in Figure 3-10 and described in detail in subsequent sections. The overall process is identical for the pickup and delivery sides. Composites are built entirely of pickup routes or entirely of delivery routes, so each set C, U, and S can be divided into disjoint sets: one for pickup and one for delivery. In the discussion that follows, we describe the procedures in terms of pickup composites and highlight any changes that occur due to delivery routes.

Single Route Composites

We begin by scanning the list of feasible aircraft routes to determine which gateway demands are covered by each aircraft route. Consider the single-leg pickup route (f,r) from gateway i to hub h flown by fleet type f. The demand b^{ih} is assumed to be either pickup or delivery. If $b^{ih} \leq u_r^f$, then $\{(f,r)\}$ is a composite cover of $\{(i,h)\}$ and is included in $S(i,h)$, the list of single route composites corresponding to gateway-hub pair (i,h). When the route has insufficient capacity, we denote as $\widehat{u}_{ih}^{fr}$ the available capacity of aircraft route (f,r) for gateway-hub pair (i,h). So if $b^{ih} > u_r^f$, (f,r) is added to the list of non-covering routes, $U(i,h)$, with available capacity $\widehat{u}_{ih}^{fr} = u_r^f$.

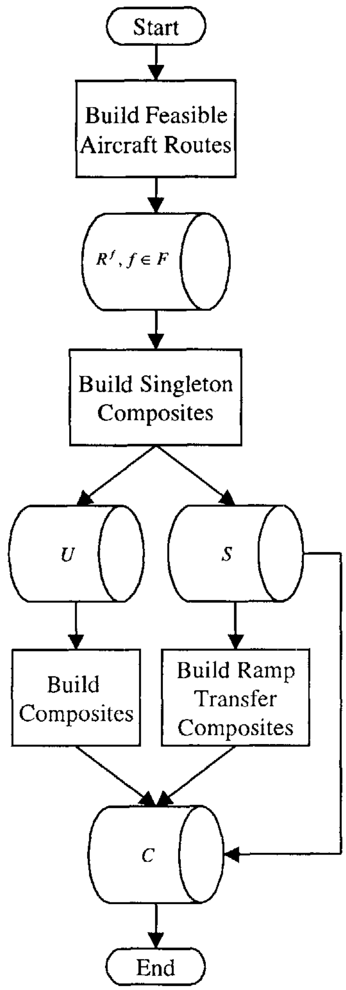

Figure 3-10: Procedure for creating composites

Consider a double-leg aircraft route (f,r) from i through j to h. If $b^{ih} + b^{jh} \leq u_r^f$, then $\{(f,r)\}$ is a composite cover of $\{(i,h),(j,h)\}$ and we store it in the single route composite lists $S(i,h)$ and $S(j,h)$. Otherwise, if $b^{ih} \leq u_r^f$, we store (f,r) in the list of non-covering routes $U(j,h)$ with available capacity $\widehat{u}_{jh}^{fr} = u_r^f - b^{ih}$; and if $b^{jh} \leq u_r^f$, we store (f,r) in the $U(i,h)$ with available capacity $\widehat{u}_{ih}^{fr} = u_r^f - b^{jh}$.

The process (shown in Figure 3-11) is repeated until the entire list of feasible aircraft routes has been scanned, for both the pickup side and the delivery side. An aircraft route may be placed in one or two single route composites lists, in one or two lists for non-covering routes, or in no list. All sets are stored by gateway-hub pair. The routes in U are used to construct multi-route composites that cover each gateway-hub demand and the routes in S are used to construct ramp transfer composites.

Multi-Route Composites

We next build the multiple route composite covers by examining each gateway-hub pair (i,h). Single-leg routes in $U(i,h)$ have their capacity dedicated to flying the demand from i to h. That is, $\widehat{u}_{ih}^{fr} = u_r^f$ and no other gateway locations would be covered by the selection of this route.

Let (f,r) be a double-leg route from i through j (or from j through i) to h contained in $U(i,h)$. According to the rules for constructing the list, (f,r) covers the demand b^{jh} but not the additional volume b^{ih}. The available capacity for (f,r) with respect to gateway-hub pair (i,h) is $\widehat{u}_{ih}^{fr} = u_r^f - b^{jh}$. (Note that if (f,r) covers b^{ih}, it cannot cover b^{jh} and would also be contained in the list $U(j,h)$.)

We build a composite $c = \{(f_1,r_1),\ \dots\ ,\ (f_k,r_k)\}$ such that $\sum_{i=1}^{k} \widehat{u}_{ih}^{fr} \geq b^{ih}$. That is, c covers (i,h). In addition, it covers all other gateways incident to the routes in c due to our method for selecting the elements of $U(i,h)$. If R_c is the set of routes included in composite c and G_c is the set of *all* gateways incident to routes $r \in R_c$, composite c covers $\mathcal{K}_c \equiv \{(g',h) : g' \in G_c\}$.

The recursive routine that generates multiple route (non-ramp transfer) composites is shown in Figure 3-12. By sorting the lists $U(g,h)$ in decreasing order of capacity, this routine generates minimal composites that cover the pair (g,h). Generating only minimal composites (which is ensured by the sorting) reduces the number of variables in the model. This may have a

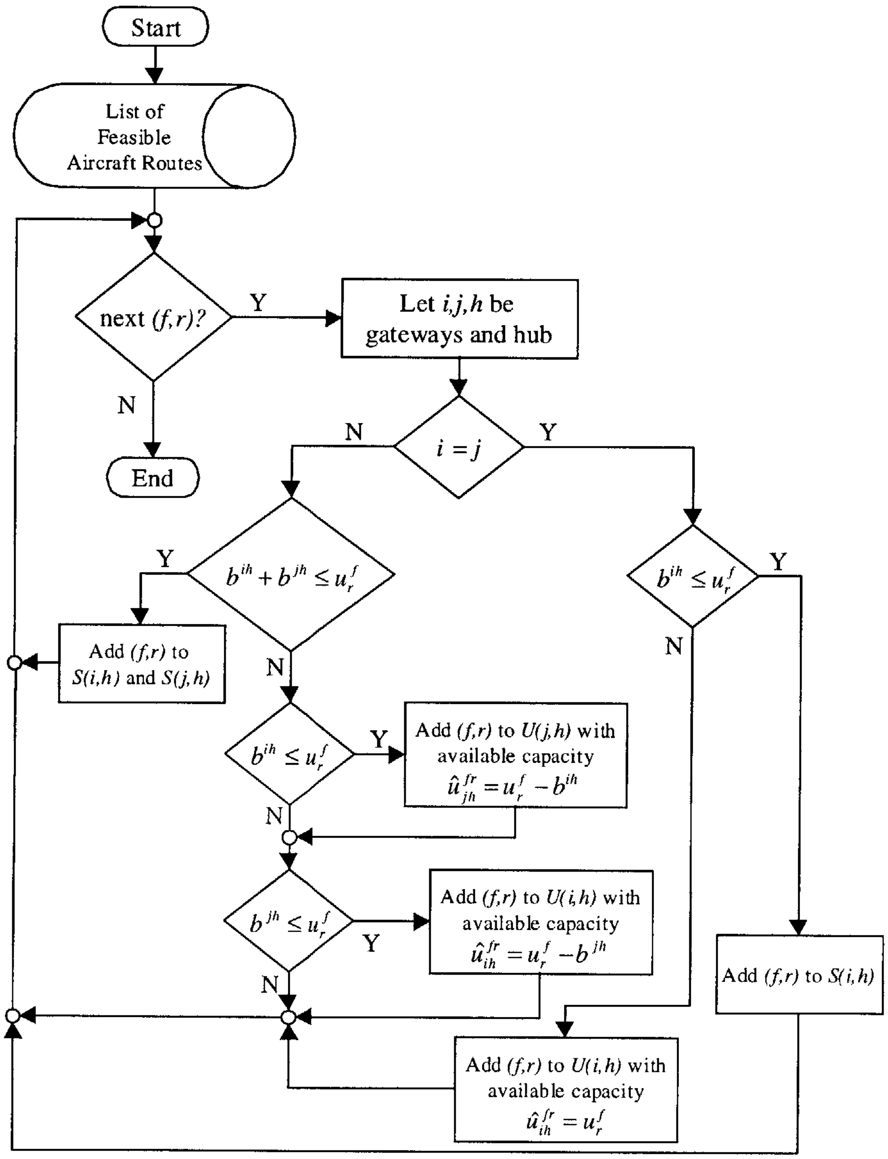

Figure 3-11: Procedure for generating single route composite list and list of aircraft routes for building multi-route (non-ramp transfer) composites

```
L ⟵ ∅
Sort U(g,h) in decreasing order of û_gh^fr
CompositeRecurse(0, b^gh, 1, L)

Procedure CompositeRecurse(capacity, demand, k, L)
    get k^th element (f,r) from U(g,h)
    while not End-of-List do
        capacity ⟵ capacity + û_gh^fr
        add (f,r) to L
        if capacity < demand then
            CompositeRecurse(capacity, b^gh, k+1, L)
        else
            Add L to C(g,h)
            Remove (f,r) from L
        end if
    end while
end Procedure
```

Figure 3-12: Recursion for creating multiple route composites, called for all gateway-hub pairs (g, h)

negative impact on our ability to attain a feasible solution with respect to the aircraft balance constraints, this is mitigated by including ferry route variables in the model.

We further reduce what are *likely* unnecessary composite variables by limiting the number of planes contained in a composite. For gateway g and hub h, let $u_{gh}^{\max}$ be the capacity of the largest aircraft that can service the pair. We limit the number of planes in each composite in $\mathcal{C}(g, h)$ to $\left\lceil \frac{b^{gh}}{u_{gh}^{\max}} \right\rceil + 1$. This is built into the recursion by a simple comparison of this value with the number of elements (k) in the composite before recursing deeper.

Finally, the composite set is further reduced by allowing only one double-leg route for a composite containing at least one single-leg route. Further, we limit composites with multiple double-leg routes to two and we do not allow those two routes to visit the same set of locations.

Example 12 *Consider the single-hub (A), single-fleet (f) network shown in Figure 3-13. All single-leg routes have capacity that covers their respective gateway-hub demands. Thus,* $S(1, A) = \{(f, 1)\}$, $S(2, A) = \{(f, 3)\}$, *and* $S(3, A) = \{(f, 5)\}$. *Next, consider the set of non-covering routes for gateway 3. This is the set of routes that do not cover* $b_P^{3,A}$ *and is defined as* $U(3, A) = \{(f, 4), (f, 6)\}$. *The available capacities for the two routes are* $\widehat{u}_{3,A}^{f,4} = 1000$ *and* $\widehat{u}_{3,A}^{f,6} =$

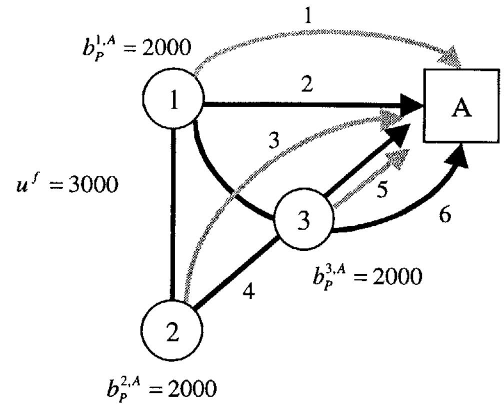

Figure 3-13: Single fleet example of building composites

1000. *Combining the two, we have* $\widehat{u}_{3,A}^{f,4} + \widehat{u}_{3,A}^{f,6} = 2000 = b_P^{3,A}$. *Therefore* $c = \{(f,4),(f,6)\}$ *is a composite cover of* $\{(1,A),(2,A),(3,A)\}$. *Route 6 is also contained in the set of non-covering routes for gateway 1, with an available capacity of* $\widehat{u}_{1,A}^{f,6} = u_6^f - b_P^{3,A} = 1000$. *It can be used (along with route 2) to build a composite that covers the gateway 1 demand* $b_P^{1,A}$.

Ramp Transfer Composites

The same underlying idea – representing multiple aircraft routes with a single variable – enables us to model ramp transfers. A ramp transfer refers to the process of moving packages from one airplane to another at an intermediate gateway location. The packages are not sorted at this location. Rather, they are simply taken from an inbound aircraft and either placed directly onto another aircraft or left on the ramp for another aircraft to pickup at a later time. The primary benefit of ramp transfers is that they allow you to reduce the number of aircraft used. Consider the following example.

Example 13 *Figure 3-14 consists of three gateways, two hubs, and a single fleet type with a capacity of 3000 packages. Without ramp transfers and allowing aircraft to be unloaded only at their terminating hub, we would require a minimum of four aircraft to cover the six gateway-hub demands. Allowing ramp transfers, we can cover the demands with two aircraft, as shown in Figure 3-14. We denote gateway-hub pairs by* (g, h). *Route 1 carries volume for (1,B), (1,A) on its first leg and (1,B), (2,B), and (3,B) on its second leg. Route 2 carries (2,A) and (2,B)*

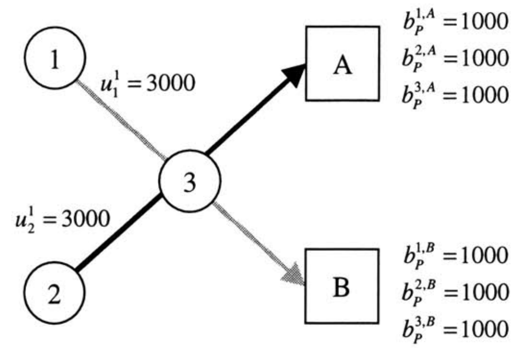

Figure 3-14: Network with ramp transfers (b_P^{gh} denotes the pickup volume for gateway-hub pair (g, h) and u_r^f denotes the capacity of fleet f flying route r)

on its first leg and (1,A), (2,A), and (3,A) on its second leg. (We have assumed that the timing of the routes is sufficient to allow the transfer of (1,A) and (2,B) volume to occur at gateway 3.)

Ramp transfer composites are subject to other operational restrictions. For instance, planning restrictions limit the transfer volume to $\frac{1}{2}$ of the capacity of the inbound aircraft and require 30 minutes between the arrival of the losing aircraft and the departure of the gaining aircraft. The flexibility of **ARM**'s composite variable construction allows these and additional operational constraints to be included as the composites are built. In the traditional network design approach (i.e., **ESSND**), this type of operational constraint would affect the model's tractability.

For each gateway g with nonzero demand for multiple hubs, we build ramp transfer composites from the single-route composites $S(g, h)$, for all h such that $b^{gh} > 0$. We join single routes from these sets such that timing restrictions and flight leg capacity limits are satisfied. Depending upon how package flows are assigned, a given composite might cover different gateway-hub demands, as demonstrated in the following example.

Example 14 *Given the composite shown in Figure 3-14 and the fact that all six gateway-hub demands are covered, the indicators used in **ARM** (constraint (3.12)) would be* $\delta_c^{1,A} = \delta_c^{1,B} = \delta_c^{2,A} = \delta_c^{2,B} = \delta_c^{3,A} = \delta_c^{3,B} = 1$. *In Figure 3-15, Route 3 is a composite cover of* $\{(4, C), (3, C)\}$. *By combining it with the ramp transfer cover in Figure 3-14, assuming the timing is adequate,*

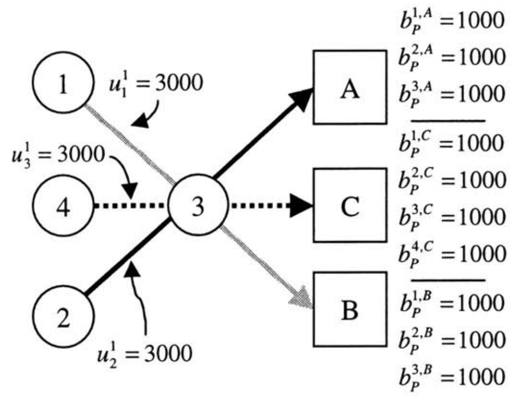

Figure 3-15: Example of a composite having multiple covers

we can transfer packages onto the second leg of route 3 at gateway 3, which has an available capacity of 1000 packages. There are two choices, however. With 1000 units of excess capacity on the first leg of route 1, we can move $b_P^{1,C}$ from gateway 1 to hub C by transferring the volume from route 1 to route 3 at gateway 3. However, no additional capacity exists to move the $b_P^{2,C}$ volume. The corresponding covering indicators are $\delta_c^{1,C} = \delta_c^{3,C} = \delta_c^{4,C} = 1$ and $\delta_c^{2,C} = 0$. If, instead, the second leg of route 3 receives a transfer of $b_P^{2,C}$ volume, the covering indicators would be $\delta_c^{2,C} = \delta_c^{3,C} = \delta_c^{4,C} = 1$ and $\delta_c^{1,C} = 0$. Thus, we see that in the case of ramp transfers, a single composite may be associated with multiple covers.

We build ramp transfer composites using the single route composite list $S(g, h)$ for each gateway g and hub h. We look for opportunities to use g as a transfer point and to utilize excess capacity on the legs of routes incident to g. We will frame the detailed discussion in terms of pickup routes, but delivery ramp transfers are constructed similarly.

Given a gateway g, let h_1 denote a hub for which $S(g, h_1) \neq \emptyset$. We select an aircraft route, $(f_1, r_1) \in S(g, h_1)$ for which g connects directly to the hub. For some other hub $h_2 \neq h_1$ such that $S(g, h_2) \neq \emptyset$, we select an aircraft route, $(f_2, r_2) \in S(g, h_2)$ and determine whether a transfer can be made between the two routes. We continue adding aircraft routes that do not share a gateway (other than g) nor a hub with routes that have already been included in the composite. This procedure is summarized in Figure 3-16.

```
L ⟵ ∅   stores the aircraft routes in the composite
B ⟵ ∅   stores the covered gateway-hub pairs
Call RampTransferRecurse(L, B, S(g, h))

Procedure RampTransferRecurse(L, B, S(g, h))
    scan first element (f, r) from S(g, h)
    while not End-of-List do
        if (f, r) has 2 legs then denote as i the gateway other than g
        if L = ∅ then
            Add (g, h) and (i, h) (if it exists) to B;  Add (f, r) to L
        else if h and i (if it exists) are not included in B then
            Scan first element (f', r') from L
            while not End-of-List do
                if feasible transfer from (f, r) to (f', r')
                  or from (f', r') to (f, r) then
                    Update available capacities on legs of (f, r) and (f', r')
                    Add (f, r) to TempL
                    Add covered gateway-hub pairs to TempB
                end if
                Scan next element (f', r') from L
            end while
        end if
        if TempL ≠ ∅ then
            L ⟵ L ∪ TempL; B ⟵ B ∪ TempB
            Add L to C(g*, h*) for (g*, h*) ∈ B
            Select h' ∉ B such that S(g, h') ≠ ∅
            RampTransferRecurse(L, B, S(g, h'))
            L ⟵ L\TempL; B ⟵ B\TempB
        end if
        scan next element (f, r) from S(g, h)
    end while
end Procedure
```

Figure 3-16: Ramp transfer composite generation procedure. This subroutine is called for all gateway-hub pairs (g, h).

The inner-most loop of this procedure determines whether ramp transfers can be made. A "feasible transfer" may occur when two basic conditions are met. First, the inbound leg and the outbound leg (which are each on different routes) must have adequate capacity. Second, the departure of the second aircraft route cannot occur earlier than a specified time after the arrival of the first (e.g., 30 minutes).

Additional operational constraints relating to ramp transfers may be inserted at this point. For instance, the planners enforce a strict rule that package volume amounting to not more than half of the inbound aircraft's *capacity* may be transferred at the intermediate location. If we wish to limit the number of aircraft routes in any ramp transfer composite, we insert a check that would prevent deeper recursion if the maximum number of aircraft routes has already been included (in $TempL$).

Drop-Off/Drop-On Routes

Additional savings might be attained by exploiting excess capacity on double-leg routes that visit multiple hubs. In the case of pickup routes, we currently treat the final location as a hub and all others as gateways. For a double-leg route (f, r) going from $1 - 2 - 3$, where both 2 and 3 are hubs, we allow only the demand $b_P^{1,2}$ and $b_P^{2,3}$ to be covered by (f, r). However, if the first leg's capacity exceeds the demand for commodities $(1, 2)$ and $(1, 3)$, that is $u_r^f > b_P^{1,2} + b_P^{1,3}$, we may use the route to pickup $(1, 2)$ and $(1, 3)$ demand at location 1, drop-off $(1, 2)$ volume at location 2, pickup $(2, 3)$ volume at location 2, and continue to the hub.

In the **ARM** implementation, we require the drop-off to cover the *entire* volume to be delivered from the originating gateway to the intermediate hub. We could broaden the composite options by allowing this aircraft route to carry a portion of the $(1, 2)$ demand, with the remaining $(1, 2)$ demand carried by capacity on some other aircraft. We would simply include (f, r) in the non-covered route list $U(1, 2)$ with an available capacity of $\widehat{u}_{1,3}^{fr} = u_r^f - b_P^{1,3}$. When multi-route composites are built (as in Figure 3-12), this route would be combined with other routes that provide partial capacity for the $(1, 2)$ demand.

3.4 Summary

Solving realistic instances of the Express Shipment Service Network Design problem via common network design formulations and solution strategies is not always possible. Instead, we present a new strategy using only design (i.e., aircraft route) variables and a reformulation approach that creates *composites* of aircraft route variables and yields an optimization model that is computationally attractive. The packages flows, previously modeled explicitly as continuous variables, are captured implicitly within our definition of the composites. Any solution to the model – which we call the Aircraft Routing Model (**ARM**) – is guaranteed to have a feasible flow.

In theory, this model will produce the same solution as the network design formulations (Chapter 2) under the assumption that commodities are specified by gateway-hub pairs. This follows from the fact that the optimal solution to the latter is, itself, a composite variable. The primary challenge, then, is to find a concise form of the composite variable formulation. That is, we must determine how we can represent any optimal solution as the combination of "smaller" composites.

In practice, this notion of smaller composites is at the heart of this reformulation strategy. By restricting the members of the composite family, C^r, we reduce the number of decision variables and, consequently, generate excellent solutions quickly. By carefully designing the methods for constructing composites, we can easily build other operational considerations into **ARM**. This is demonstrated in the next chapter's computational case study.

Chapter 4

Case Study, Computational Results, and Analysis

In this chapter, we apply the Aircraft Routing Model (**ARM**) to the Next-Day Air (NDA) network of the United Parcel Service (UPS). We examine **ARM**'s computational performance in three areas. First, we establish the trade-off between the complexity of the composite variables and the quality of the resulting model. Increasing the number of composite variables typically improves solution quality but slows the model's run-time. Second, we compare **ARM**'s solution relative to the manual solution generated by the carrier's planners. We examine both aggregate measures and specific differences in the model's solution versus the planners' solution. Finally, we use the model to explore radical, strategic changes to the air network, including the case in which the carrier has complete freedom in selecting a fleet mix and the case in which the carrier operates a single mega-hub with no additional hubs. This underscores **ARM**'s ability to function in a broader spectrum of strategic planning, not just in the development of aircraft routes.

4.1 System Description

In Chapter 2, we presented a general overview of the Express Shipment Service Network Design problem. In Chapter 3, we added to that description the assumption that the package demands were specified by gateway-hub pairs. Further, we described modifications to the constraints that would allow **ARM** to be effective in practice. Here, we apply **ARM** to the UPS-specific setting, selecting constraints and building composites consistent with the carrier's operation.

Long-range aircraft routing plans are developed annually against a single set of demand estimates. These plans specify aircraft movements for ten months of a given year. Because the system is driven by customer requirements, the demand estimates are conservative, erring on the side of overestimating demand. A different set of data is used for the peak retail season that occurs during November and December. Considering other demand scenarios is currently not possible for the planner due to the considerable amount of time involved in manually creating aircraft plans.

In the general **ESSND** problem, demands are specified by origin gateway and destination gateway. These demands arise from the compression of customer-to-customer demands into the air gateways. That is, local demands are assigned to be handled at specific gateways (it is assumed that ground vehicles and small feeder aircraft will provide the capacity to move demands from customer centers to air gateways). The result is a set of demands specified by origin gateway and destination gateway and these demands are input to the **ESSND** problem.

With **ARM**, we assume that this demand compression goes one step further. Planners specify *service territories* for each hub. In addition, they determine which demands will be handled at each hub. The result is that demands may now be specified as gateway-to-hub volume on the pickup side and hub-to-gateway volume on the delivery side. These demands are input to **ARM**.

Demands are measured in either packages or containers. The planning scenario used throughout this chapter consists of a nightly volume of 2250 containers carrying 926268 packages on the pickup side and 2288 containers carrying 967172 packages on the delivery side. When planning with packages, a model that generates fractional packages is acceptable, since fractional packages can be easily absorbed into the capacity. When working with containers, the model must enforce the integrality of flows – as **ARM** does – because a fraction of a container might represent a significant portion of aircraft capacity. Planners typically work with containers as they provide a more realistic characterization of the demand's footprint on an aircraft. Consequently, all scenarios presented in this chapter are based on container demands.

The air network consists of 101 locations, seven of which are hubs. Distances between locations are measured in *block hours* (i.e., flying time plus taxi time) that are dependent upon fleet type. *Level-of-service* requirements (i.e., Early Pickup Times and Late Delivery Times) are specified for each gateway location. Each hub has a *sort start* and a *sort end* time, which may be route-specific at each hub. For example, pickup routes from west coast locations to non-west coast hubs may arrive later than routes originating from non-west coast locations. Likewise, delivery routes from a non-west coast hub to a west coast gateway may depart the hub earlier than a route destined for non-west coast locations.

The carrier's fleet includes seven aircraft types and 160 total aircraft. Each fleet type has an associated flying range and a set of locations and hubs at which it may land. The capacity of each fleet type is specified in terms of containers. A fleet type's capacity is constant for all routes except for pickup routes going from west coast gateways to non-west coast hubs, in which case the plane's capacity is reduced by 30 percent to compensate for tighter time windows associated with receiving and sorting west coast volume. (This capacity reduction is in place only for planning purposes.)

In this chapter, we do not specify the carrier's actual cost of owning and operating aircraft to ensure the confidentiality of their cost data. When we examine model performance as a function of input parameters, we use industry standard costs to solve **ARM** and to make comparisons between runs. When comparing with the planners' solution and in performing scenario analyses, we run **ARM** with company costs and present comparisons based on percentages.

All computations were performed on an HP9000 Model D370 running HPUX 10.20. All models were compiled using HP's ANSI C/C++ compiler with calls to the ILOG CPLEX 6.5 Callable Library [46]. CPLEX MIP optimizer settings, unless otherwise noted, are shown in Table 4.1. For parameters not described, the CPLEX default values were used.

4.2 Computational Effect of Composite Definition

Applying **ARM** to the UPS problem, we restrict composites to those that satisfy conditions resulting from legitimate operational constraints or from the modeler's desire to reduce problem size. How we define this set affects both solution quality and computational effort. The 13,936 feasible aircraft routes, generated based on the considerations identified in the previous section,

Parameter	Setting
Root node algorithm	Barrier with crossover
Subproblem algorithm	Dual simplex
Branching direction	Up direction selected first
Node selection	Best estimate search
Variable selection	Base on pseudo reduced costs
Subproblem pricing	Dual steepest edge pricing
Root node heuristic	Turned on
Heuristic frequency in B&B	Every 100 nodes

Table 4.1: Settings for CPLEX 6.5 MIP solver

		Baseline
Composites	Single Route	7237
	Mult-Route	24078
	Ramp Transfer	0
	Ferry Routes	0
Problem Size	Columns	26525
	Rows	909
	Nonzeroes	205170
Objective Value ($M)	LP Relaxation	1.65120
	First Integer	1.65906
	Optimal Integer	1.65434
	LP-IP Gap	0.0019
Run Time (sec.)	LP Relaxation	27.23
	Optimal Integer	1188

Table 4.2: Computational results of baseline **ARM** solution

serve as the building blocks for our composites.

We begin with a baseline solution that includes neither ramp transfer composites nor ferry variables. It enforces aircraft balance at all locations and allows *quasi-balance* (see Chapter 3) in the LA Basin and in northern California. The computational results are shown in Table 4.2. The first section shows how many of each composite type are built. The second shows the problem size after standard CPLEX preprocessing (see ILOG [46]). The third section shows solution cost (in millions of dollars per night) as well as the LP-IP gap, which is the difference of the optimal integer and LP values, relative to the LP value. The final section shows the running time of the LP relaxation and the time taken to find the optimal integer solution.

		Maximum Ferry Route Length (block hours)					
		Baseline	1.0	1.5	2.0	2.5	3.0
Composites	Single Route	7237	7237	7237	7237	7237	7237
	Mult-Route	24078	24078	24078	24078	24078	24078
	Ramp Transfer	0	0	0	0	0	0
	Ferry Routes	0	3653	9700	17404	24895	30957
Problem Size	Columns	26525	32777	38838	46541	54032	60094
	Rows	909	1057	1064	1064	1064	1071
	Nonzeroes	205170	231968	244090	259496	274478	286602
Objective Value ($M)	LP Relaxation	1.65120	1.65075	1.65048	1.64976	1.64932	1.64928
	First Integer	1.65906	1.66395	1.65518	1.65968	1.65836	1.66531
	Optimal Integer	1.65434	1.65413	1.65351	1.65217	1.65224	1.65224
	LP-IP Gap	0.0019	0.00205	0.0018	0.0014	0.0018	0.0018
Run Time (sec.)	LP Relaxation	27.23	35.32	44.66	52.77	63.79	68.53
	Optimal Integer	1188	2041	2320	3165	7650	9214

Table 4.3: **ARM** solution varying maximum ferry distance (distance parameter is block hours)

4.2.1 Ferry Route Length

Ferry routes are used to reposition aircraft so that the system begins the next cycle of the Next-Day Air (NDA) system in the same state it began the current cycle. Ferry routes between gateways and hubs include any feasible aircraft routes found in the route generation procedure. This includes all single-leg aircraft routes, whether or not they cover their gateway-hub demands. In addition, we consider direct gateway-to-gateway ferry routes that are shorter than some specified length (measured in block hours).

We examine the effect of gateway-to-gateway ferry route length on **ARM**'s size and its solution quality (see Table 4.3). Increasing the maximum ferry route length slightly decreases the optimal objective function value. Relative to the baseline solution, ferry routes do not appear to add significant cost savings. They do, however, provide a means for generating a solution that satisfies the aircraft balance constraint and that might not otherwise exist. Furthermore, when we include ramp transfer composites (in the next section), ferry routes provide additional flexibility that allows integer solutions to be generated more quickly.

		Maximum Ramp Transfer Percentage				
		0.00	0.05	0.15	0.25	0.50
Composites	Single Route	7237	7237	7237	7237	7237
	Mult-Route	24078	24078	24078	24078	24078
	Ramp Transfer	0	1714	12287	21581	27146
	Ferry Routes	9700	9700	9700	9700	9700
Problem Size	Columns	38838	42625	53198	62492	68057
	Rows	1064	1064	1064	1064	1064
	Nonzeroes	244090	281470	397484	497531	557474
Objective Value ($M)	LP Relaxation	1.65048	1.64901	1.62357	1.60261	1.57949
	First Integer	1.65518	1.65952	1.64008	1.61478	1.62377
	Best Integer	1.65351	1.65302	1.62766	1.61478	1.59433
	LP-IP Gap	0.0018	0.0024	0.0025	0.0075	0.0094
	Best Lower Bound	1.65351	1.65302	1.62766	1.60934	1.58610
	Gap	0.0000	0.0000	0.0000	0.0034	0.0052
Run Time (sec.)	LP Relaxation	44.66	49.34	49.38	64.86	71.82
	Best IP	2320	2490	4928	29118	75771

Table 4.4: **ARM** solution varying maximum ramp transfer load (parameter is ratio of maximum ramp tranfer load to the inbound aircraft capacity)

4.2.2 Maximum Ramp Transfer Load

We turn to the case when we include ramp transfer composites. One parameter that controls how these composites are built is the maximum volume allowed to be ramp transferred. This limit is determined as a percentage of the capacity of the *inbound* aircraft and, consistent with operational planning rules, is set to 50 percent. We examine the effect of setting this percentage at different levels: 5%, 15%, 25%, and 50%. The higher percentage yields more transfer options and increases the number of composite variables.

The results for the maximum transfer load are shown in Table 4.4. These runs were made only allowing ramp transfers that involve two aircraft. We report the best lower bound established during branch-and-bound and the gap between this bound and the best integer solution (relative to the best bound). As the number and complexity of composites increase, finding the optimal integer solution becomes more difficult. As shown in Figure 4-1, the gap between the best integer solution and the tightest lower bound increases as the load approaches the maximum of 50 percent. But this gap is still small – 0.52 percent. In addition, the difference between the LP relaxation and the tightest lower bound increases as we increase the

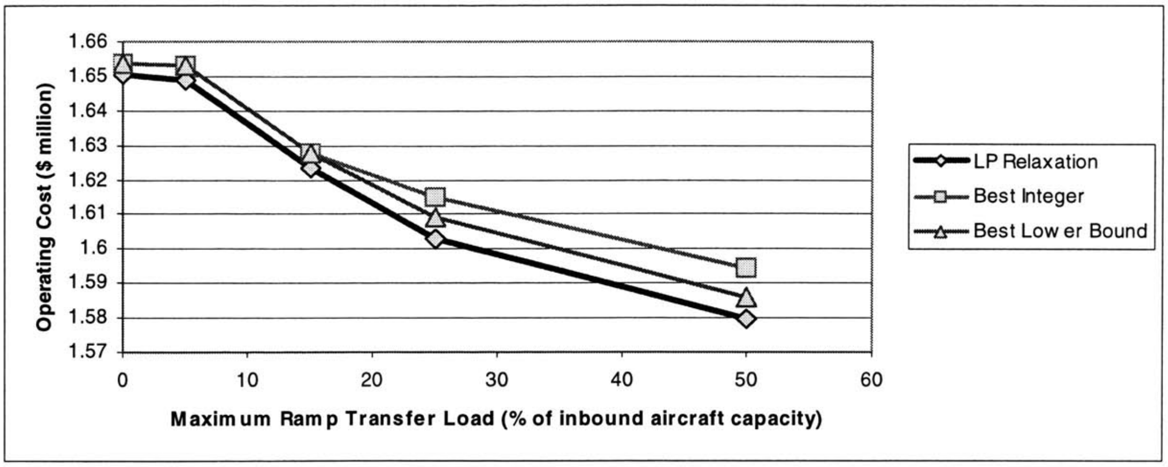

Figure 4-1: Objective function and bounds versus maximum ramp transfer load

number of composite variables, suggesting that as the complexity of the ramp transfer composites increases, the lower bound given by the LP relaxation weakens. This weakening is slight, however, as the worst LP-IP gap is still 0.94 percent, which was found short of optimality.

4.2.3 Number of Routes per Ramp Transfer Composite

We next examine the effect of the complexity of our ramp transfer composites on solution quality and run-time. Recall from Chapter 3 that for each gateway-hub pair (g, h) involved in a ramp transfer, our method of construction requires that the (g, h) demand be assigned to a single aircraft on the inbound side and a single aircraft on the outbound side. This follows from the planning consideration that if gateway-hub volume is to be ramp transferred it cannot be split, neither at its origin nor at its ramp transfer gateway.

This restriction implies an upper bound on the number of aircraft contained in a ramp transfer composite. For example, if the gateway belongs to two hub territories, a ramp transfer composite may contain at most two aircraft routes. For a ramp transfer gateway belonging to three hub territories, that number is three. For gateways belonging to four hub territories (there are only four such gateways), the number is four. Thus, there is a natural limit on the complexity of ramp transfer composites based on the number of hub territories in which the gateway lies.

		Maximum Number of Aircraft Routes in RT			
		0	2	3	4
Composites	Single Route	7237	7237	7237	7237
	Mult-Route	24078	24078	24078	24078
	Ramp Transfer	0	12287	28462	36092
	Ferry Routes	9700	9700	9700	9700
Problem Size	Columns	38838	53198	69373	77003
	Rows	1064	1064	1064	1064
	Nonzeroes	244090	397484	667533	827064
Objective Value ($M)	LP Relaxation	1.65048	1.62357	1.62158	1.62158
	First Integer	1.65518	1.64008	1.63468	1.63601
	Optimal Integer	1.65351	1.62766	1.62586	1.62586
	LP-IP Gap	0.0018	0.0025	0.0026	0.0026
Run Time (sec.)	LP Relaxation	44.66	49.38	102.29	105.94
	Optimal Integer	2320	4928	11817	6753

Table 4.5: **ARM** solution varying maximum number of aircraft routes allowed in ramp transfer composites

In addition to this natural limit, we may place an arbitrary limit on the number of routes in any ramp transfer composite. We explore the effect of this limit on problem size and solution quality. As in the baseline case, we include ferry routes with a maximum length of 1.5 block hours to provide **ARM** more options to find a balanced solution. We set the maximum ramp transfer load to 15 percent of the inbound aircraft route's capacity.

Table 4.5 shows the effect of the complexity of composites upon solution quality and run time. We vary the maximum number of aircraft routes allowed in each ramp transfer composite. The objective function value of the best integer solution is reduced by a total of 1.56 percent when we introduce the two-route ramp transfer composites and by a total of 1.67 when we use three-route ramp transfer composites. Note that the addition of four-route ramp transfers improves neither the LP relaxation nor the best lower bound. The magnitude of these improvements will grow when we increase the maximum ramp transfer load from 15 percent to its upper limit of 50 percent (as shown in Table 4.4). Finally, notice the behavior of the LP-IP gap. When ramp transfers are included, the gap jumps from 0.18 percent to 0.25 percent but building more complex ramp transfer variables has little effect on the gap.

4.2.4 Effect of Aircraft Balance Constraints

As a final comparison, we examine the effect of the aircraft balance constraints. Using the same composites as in the baseline case, we run **ARM** without the gateway balance requirement. We continue to enforce balance at the hubs; that is, the number of planes of a given aircraft type arriving to a hub's sort, including ferry routes, must be offset by the same number of planes of that type departing the hub following the sort, including ferry route departures. Not only does the removal of the gateway balance requirement reduce the objective value by 4.59%, but the run time is reduced dramatically. While the root node run-time is reduced by less than half, the "unbalanced" model finds the optimal solution in roughly one-tenth the time it takes to find the optimal integer solution of the baseline model.

This comparison yields three key observations. First, balancing the design variables is detrimental to running time. Second, without balance, the LP relaxation provides a much tighter bound on the optimal integer solution. Third, without balance, **ARM** produces a solution with significantly lower cost (almost $76K per day, or a reduction of 4.59%) for a model without ramp transfer composites. If aircraft used in the NDA network can be repositioned through clever design of the Second-Day Air (SDA) network, we might be able to realize a portion of these operating cost savings in the NDA network. Thus, a natural, and likely profitable, extension to **ARM** is its application to the combined NDA/SDA network design problem.

4.3 ARM Solution Versus Planners' Solution

We now compare the **ARM** solution directly with the UPS planners' solution. Based on the computational trade-offs discovered in the previous section, we generate composites using a maximum of three-route ramp transfers, ferry routes with a maximum length of 1.5 block hours, and maximum ramp transfer loads of half the capacity of the inbound aircraft. The planners' solution attempts to minimize operating cost so we mirror that objective with **ARM**. The costs we use are carrier-specific (versus industry standard). Furthermore, we treat the SDA network as fixed. This establishes boundary conditions that our NDA solution must satisfy.

		Baseline	No Gateway Balance
Composites	Single Route	7237	7237
	Mult-Route	24078	24078
	Ramp Transfer	0	0
	Ferry Routes	0	0
Problem Size	Columns	26525	29216
	Rows	909	434
	Nonzeroes	205170	174208
Objective Value ($M)	LP Relaxation	1.65120	1.57725
	First Integer	1.65906	1.58839
	Optimal Integer	1.65434	1.57843
	LP-IP Gap	0.0019	0.00075
Run Time (sec.)	LP Relaxation	27.23	18.46
	Optimal Integer	1188	121.09

Table 4.6: **ARM** solution with and without gateway balance

4.3.1 ARM Solution Minimizing Operating Cost

We run **ARM** using the objective to minimize operating cost, which consists of two components. First, an aircraft incurs a fixed cycle cost for each leg it flies. Second, an aircraft incurs a variable cost based on the block hours it flies. Combining these elements, we are able to determine the operating cost of an aircraft route because both the number of legs and the length of those legs are known *a priori*.

With the nightly operating cost of the Next-Day Air network in the millions of dollars, each percentage point saved translates to significant savings (see Table 4.7). The 6.96 percent reduction in operating cost translates to over 20 million dollars in annual savings. The more significant savings, however, come from the reduction in the number of aircraft and the corresponding reduction in daily aircraft ownership cost (with a new aircraft costing in the tens of millions of dollars).

The gaps shown in the table are, for "Best Bound," the difference between the best integer solution and the best lower bound, relative to the best lower bound; and for "LP-IP," the difference between the best integer solution and the LP relaxation, relative to the LP relaxation. In the branch-and-bound tree, the best lower bound does not improve significantly, which is why the gaps are similar. The running time for **ARM** to obtain the 5^{th} integer solution, which is the solution we report in the table, was just over 100 minutes.

		ARM
Problem Size	Columns	124572
	Rows	1117
	Nonzeroes	1492014
Solution (% improvement from carrier's solution)	Operating Cost	6.96%
	Cycle	4.74%
	Hourly	8.22%
	Number of aircraft	10.74%
	Aircraft Ownership Cost	29.24%
	Total Cost	24.45%
Run Time (sec.)	LP Relaxation	317.10
	Best IP	6324
Optimality Gaps	Best Bound	2.14%
	LP-IP	2.14%

Table 4.7: **ARM** versus planners' solution, with objective to minimize operating cost

A comparison of fleet type usage is shown in Figure 4-2. The left column shows available aircraft, the center column shows aircraft usage in the planners' solution, and the right column shows the aircraft usage in the **ARM** solution. The striking difference occurs with fleet types 2 and 8. **ARM** avoids flying these aircraft because they are older and have higher operating cost.

For planning with a *fixed aircraft inventory*, a large impact comes when the network is re-designed for the peak retail season in November and December. More effective use of the carrier's inventory may reduce or eliminate the need to lease additional aircraft. For long-range planning, the ability to fly with fewer aircraft may allow the purchase of fewer new aircraft or avoid the need to accelerate production on existing contracts. With a single aircraft costing tens of millions of dollars – take UPS's recent purchase of 30 Airbus A300-600 aircraft for $5 billion – avoiding or deferring the cost of a single aircraft yields significant savings.

To achieve these cost reductions, **ARM** takes advantage of aircraft capacities and gateway/hub timing restrictions. With differences in gateway-hub demands on the pickup side and the delivery side, **ARM** attempts to closely match capacity to demand while satisfying aircraft balance. Consider the routes for a given fleet type shown in Figure 4-3. Planners' solutions, for almost every route, assign fleet types to the same routes on both the pickup and delivery sides. **ARM** does not create such symmetry. Rather, it assigns this fleet type such that capacity and demand are closely matched. When we force **ARM** to obey the SDA boundary

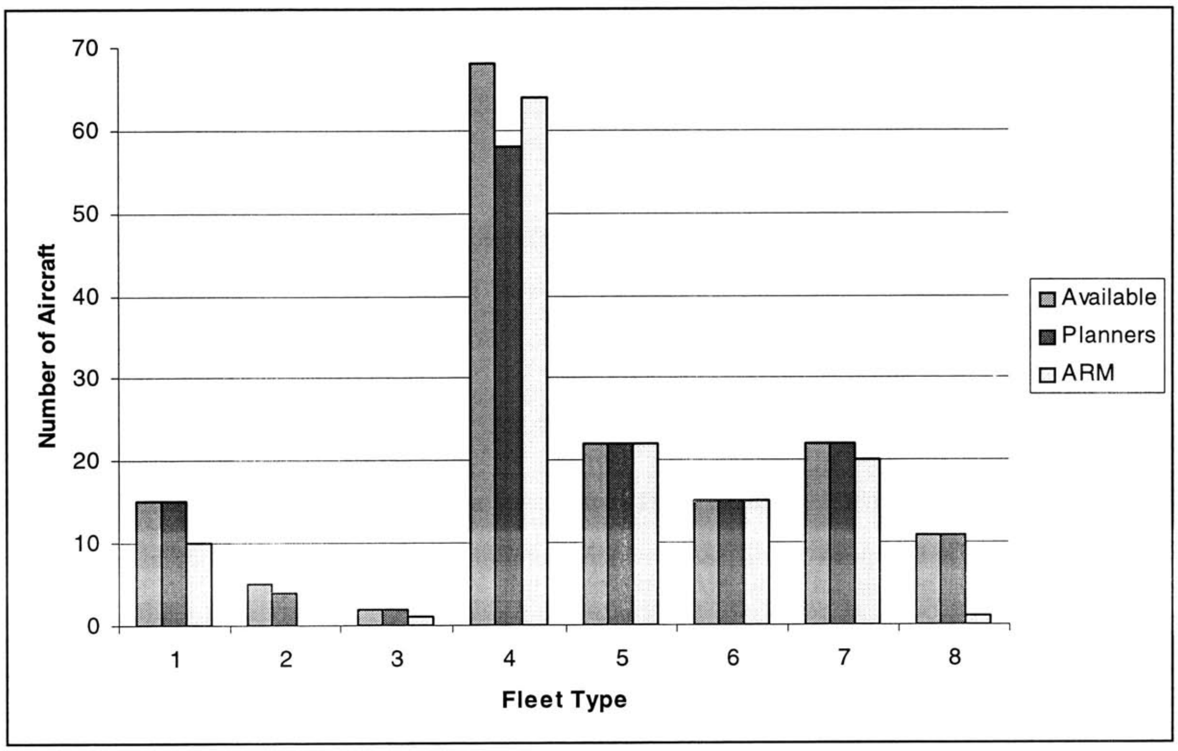

Figure 4-2: Number of aircraft used in minimum operating cost **ARM** solution

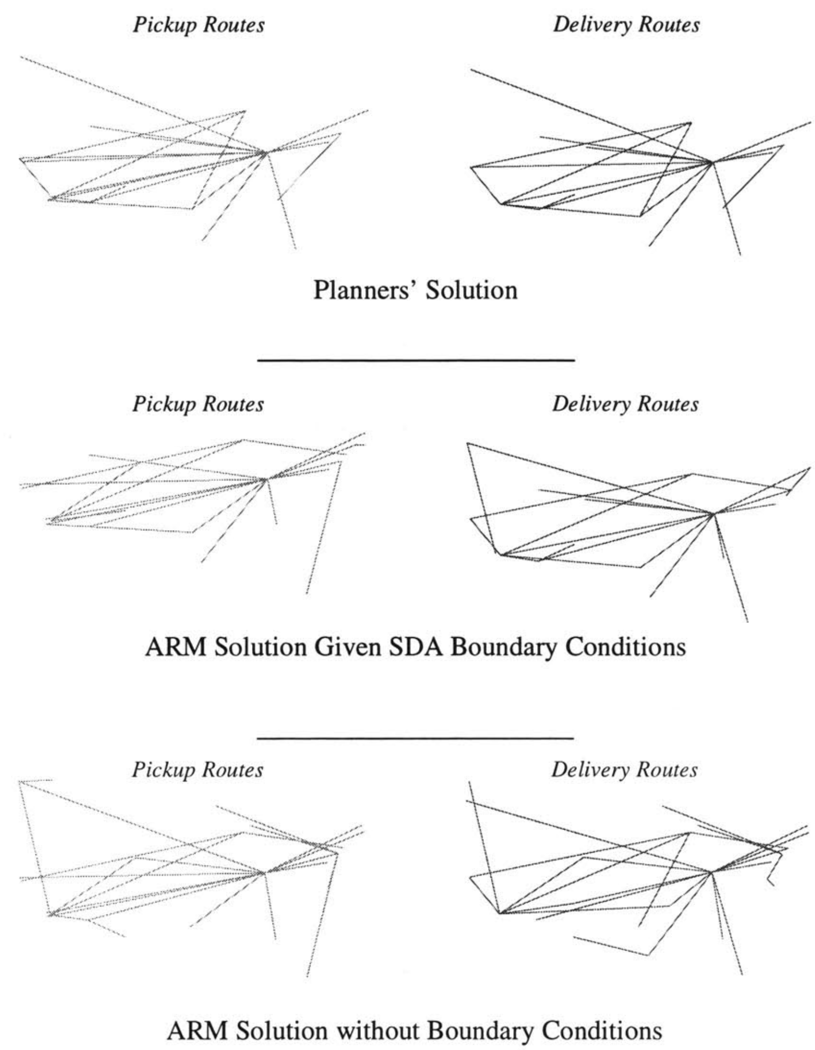

Figure 4-3: Comparison of route selection for a single fleet type

conditions, some aircraft routes are locked into routes similar to those in the planners' solution. Relaxing the SDA boundary conditions, we see a greater difference in the gateways that are serviced in **ARM**'s solution compared with the planners' solution.

Next, we see the same behavior for the aircraft routes incident to a single hub (Figure 4-4). The manual solution, again, has symmetry between the pickup and delivery sides (not shown is the symmetry by *fleet type*). The **ARM** routes, once again, are asymmetric. Although some locations appear not to satisfy aircraft balance, the balance requirement is actually satisfied by an arrival from (or departure to) another hub. Note that gateway locations in this hub's territory that are not incident to any route are serviced via ramp transfer. Their volume is picked up by an aircraft bound for a different hub and transferred to one of the routes shown in Figure 4-4.

We examine more closely the aircraft routes assigned between this hub and one of its gateways (see Figure 4-5). The planners specify a double-leg pickup route of type 4 and the reverse double-leg route for delivery. **ARM** flies a different pair of routes, taking advantage of extra capacity on the pickup route to visit an intermediate location with higher demand than the intermediate location chosen by the planners. The choice of fleet type for the terminating gateway is dictated by the SDA boundary conditions. One plane of type 4 is required to start and finish at this gateway. Without these boundary conditions, **ARM** selects a smaller fleet type and services the gateway with direct routes both on pickup and delivery.

With the objective to minimize operating cost, **ARM** assigns fleet types to routes in a manner consistent with their cycle and hourly cost. For example, a small fleet type with low cycle cost and low hourly cost is a good candidate to fly long routes. It would also tend toward double-leg routes, but may be limited in its double-leg opportunities due to its small capacity. A large aircraft with low cycle cost would not face this problem. If its hourly cost is high, however, it would be a good choice for short double-leg routes. In Figure 4-3, the fleet type shown has high capacity, low hourly cost, and low cycle cost. Only one of its pickup routes and two of its delivery routes are double-leg routes. The element preventing more double-leg routes for this aircraft type is the time it takes to turn the aircraft between legs. With a large turn time, we expect double-leg routes for this fleet type to be short and single-leg routes to be long, which is precisely what happens. The average route length (block hours, not including

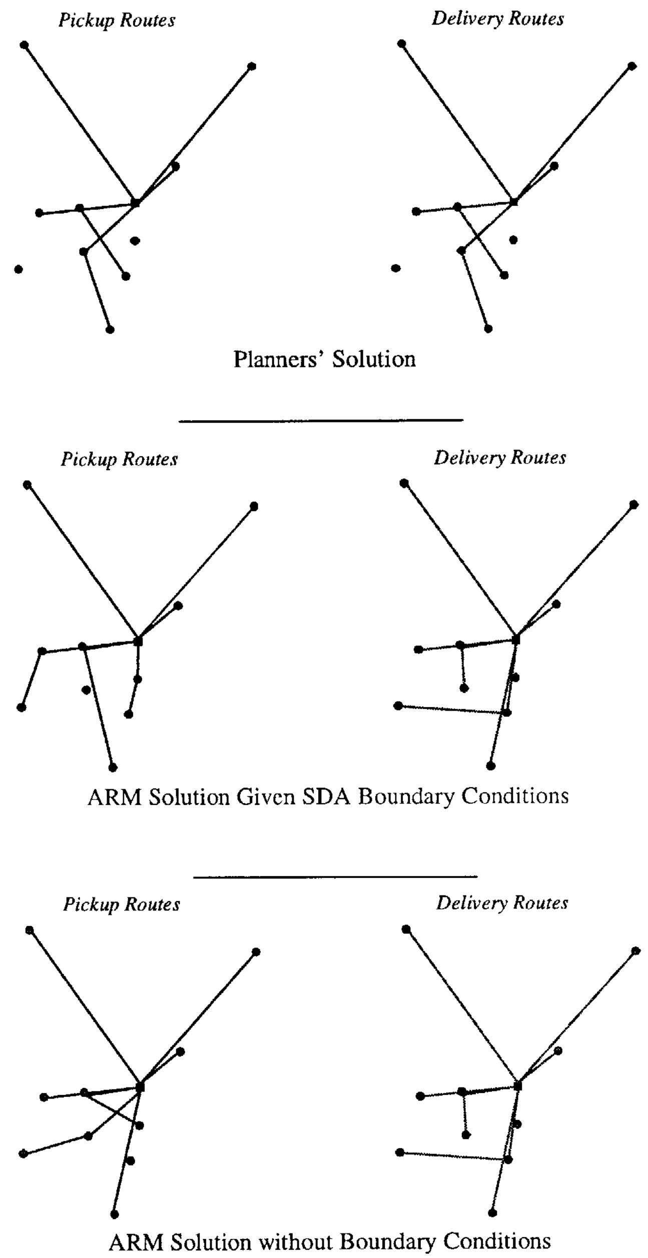

Figure 4-4: Comparison of route selection for a single hub

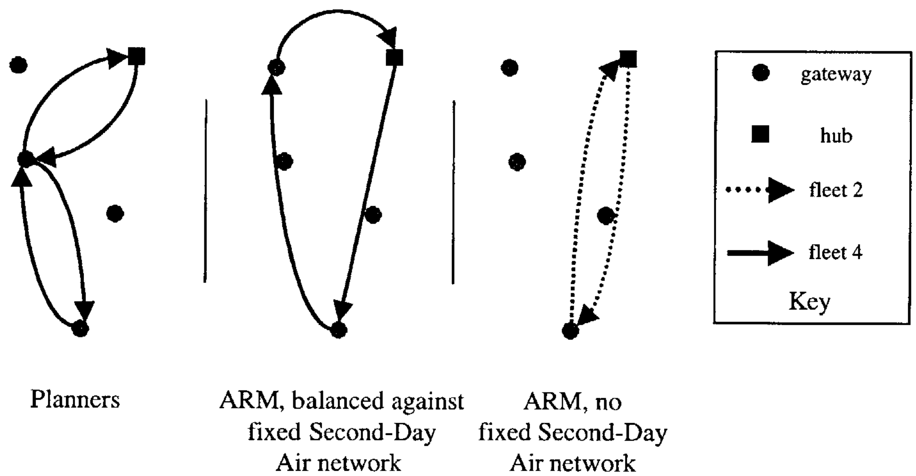

Figure 4-5: Comparison of routes incident to a single gateway-hub pair

turn time) for this fleet type is 2.56 hours, compared to 2.14 hours for the overall average route length. The planners' solution has similar usage rates.

Next, we show an aircraft type whose usage differs significantly from the planners' solution. Figure 4-6, shows the routes for fleet type 1, which has low operating cost and low cycle cost. Working against it are its short flying range (2000 miles) and its low capacity. While the low cycle cost encourages double-leg routes, there are not many pairs of gateways whose demands will both fit on this fleet type. Nonetheless, **ARM** increases the number of double-leg routes from zero to four. Furthermore, it increases the block hours per route from 1.32 hours per route to 1.63 hours per route.

Additional insight is found in the aggregate aircraft utilization (see Table 4.8). With almost eleven percent fewer aircraft, the **ARM** plan flies significantly more legs per plane and more hours per plane. The *total* block hours and *total* number of legs flown are lower in the **ARM** solution.

Finally, **ARM** selects non-intuitive routes that may not be natural for human planners to consider. Some examples are shown in Figure 4-7. **ARM** exploits both timing (Early Pickup Times and Late Delivery Times) at the gateways and utilizes available aircraft capacity by double-stopping the aircraft. The A-3-1 route is unusual in the sense that the plane flies east before flying a long leg to the West Coast. Taking advantage of time zones, the aircraft arrives

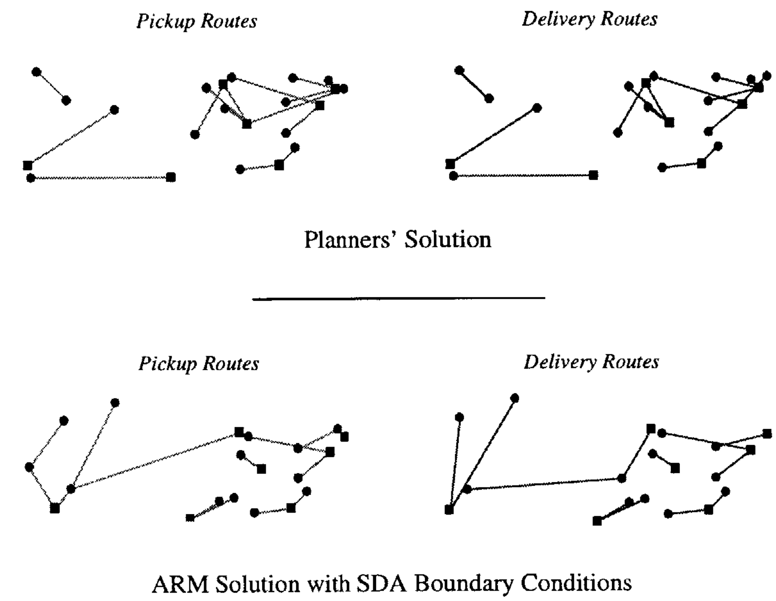

Figure 4-6: Planners' solution versus model solution for fleet type 1

		Hours	Hours/plane	Legs	Legs/plane
Pickup	**ARM**	285.92	2.15	173	1.30
	Planners	297.50	2.00	173	1.161
Delivery	**ARM**	279.49	2.13	164	1.25
	Planners	296.25	2.02	174	1.18
Total	**ARM**	565.41	4.25	342	2.53
	Planners	593.75	3.98	347	2.33

Table 4.8: Summary of plane utilization in terms of legs and distance flown

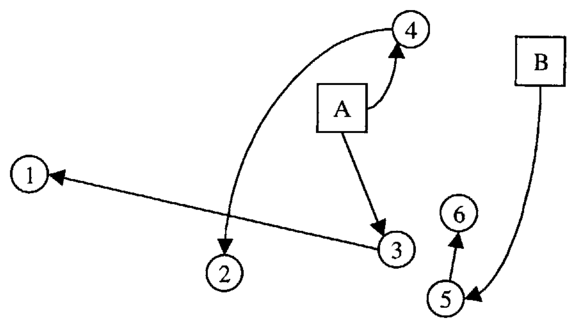

Figure 4-7: Nonintuitive double-leg routes selected by **ARM**

at gateway 1 by its LDT. Next, we have A-4-2, where the first leg is in the *opposite* direction from the second leg. In addition to flying from one time zone to the next, this route is involved in a ramp transfer at gateway 4. Thus, additional cost that may result from flying in the wrong direction is saved by not having to fly extra routes to cover the demand that is ramp transferred at gateway 4. Finally, route B-5-6 takes advantage of a late LDT into gateway 6 and participates in a ramp transfer at gateway 5.

We have provided some of the intuition behind **ARM**'s route selection. With all the planning inputs to consider – level of service requirements, hub sort times, aircraft range and speed, landing requirements at gateways and hubs, plane capacities, fleet costs, and gateway-hub demands – manually creating a solution to this massive problem is difficult and time consuming. We have demonstrated **ARM** to be an effective tool to generate a network design that performs very well with respect to operating cost. While operating costs are significant, enormous savings will result from the reduction of aircraft required to handle the demand.

4.3.2 ARM Solution Minimizing Operating and Ownership Cost

With a focus on saving aircraft, we now add ownership cost to the objective function. We assign half of the daily ownership cost on pickup routes and half on the delivery routes. Each pickup route is balanced by a delivery route, so each plane (which flies one pickup, one delivery route, and possibly one ferry route) incurs the full daily ownership cost. In addition, we incur full daily ownership cost for SDA aircraft not used in the NDA solution.

Ownership cost dominates operating cost and the resulting solutions typically have poor

		ARM
Solution (% improvement versus carrier's solution)	Operating Cost	-8.10%
	Cycle	-11.61%
	Hourly	-6.11%
	Number of aircraft	9.56%
	Aircraft Ownership Cost	37.92%
	Total Cost	28.04%
Run Time (sec.)	LP Relaxation	322.00
	Best IP	8878
Optimality Gaps	Best Bound	2.54%
	LP-IP	2.56%

Table 4.9: **ARM** versus planners' solution with objective to minimize operating plus ownership cost

operating cost relative to the planners' solution (see Table 4.9). However, the savings in aircraft ownership cost is enormous. Figure 4-8 shows aircraft usage (right column) relative to available aircraft (left column). While **ARM** uses the same number of planes as when minimizing operating cost, the choice of which planes to fly has shifted. The figure also shows, for each fleet type, the normalized ratio of its ownership cost to its capacity. The under-utilized aircraft are those with the high cost/capacity ratio. The usage of fleet types under the new objective function results from the fact that the fleet types with low ownership cost tend to be older and have higher operating cost. This is why they were not fully utilized when the objective was to minimize operating cost.

The only exception to the model's selection of aircraft based on this ratio is with the type 3 aircraft. While its ownership cost is low (relative to its capacity), both its cycle cost and hourly operating cost are more than *twice* those of any other aircraft type. Our objective function is to minimize ownership cost **plus** operating cost, and this fleet type is not appealing. The single type 3 aircraft is selected due to the requirement to balance against the existing Second-Day Air network.

4.3.3 Aircraft Arrivals at Hubs

On the pickup side, single-leg routes and the second leg of double-leg routes are constructed based on arriving at their respective hub *sort start times*. The same applies to the delivery side, with all routes departing at their hub's sort end time. If the output of **ARM** is interpreted

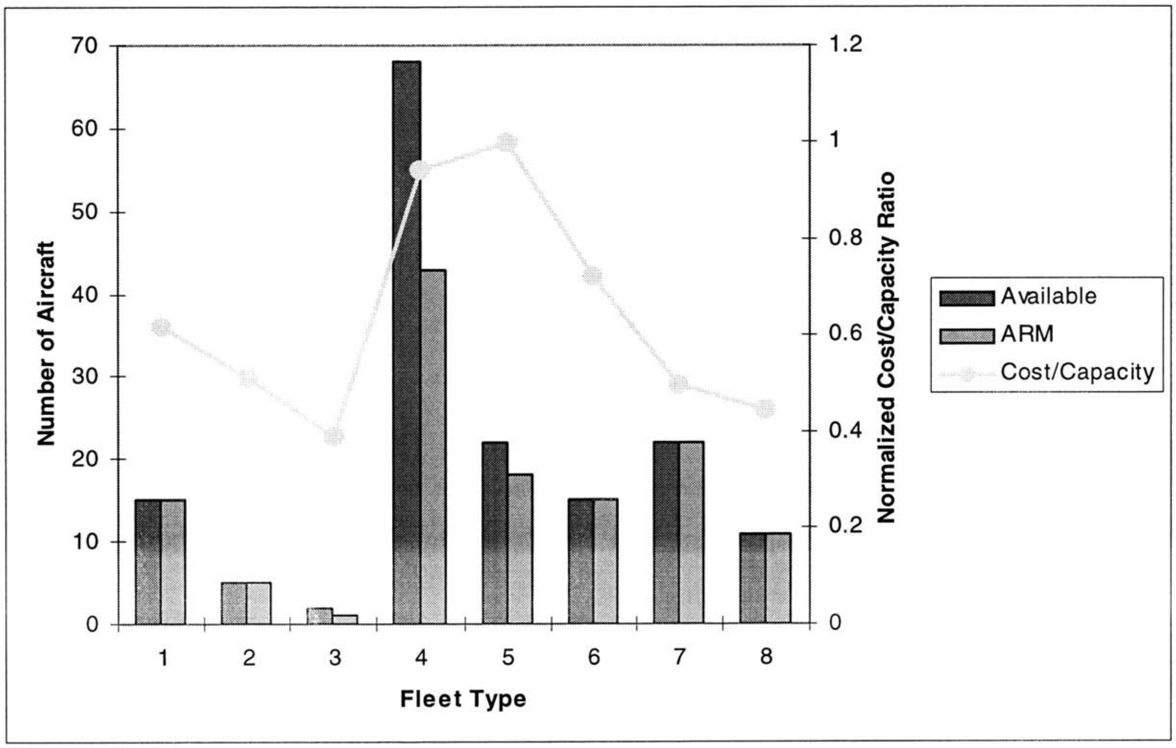

Figure 4-8: Number of aircraft used in **ARM** solution when minimizing operating and ownership cost

as a schedule (which it should not be), one would see arrivals into a given hub occurring at the same time (or, if the hub has both a west coast sort and a non-west coast sort, arrivals at two times). To build a valid schedule, these arrivals and departures must be staggered.

Creating staggered arrivals can be handled in one of two ways. First, we can analyze the current solution and determine whether such an arrival profile can be created. Second, we can introduce a set of dynamic constraints that force **ARM** to stagger arrivals into the hubs. This type of dynamic constraint may be used either to guarantee the appropriate *landing rate* or to ensure that the arrival of *package volume* is spread over the time prior to sort start.

Analysis of Current Solution

We examine the **ARM** solution found by optimizing with respect to operating cost. For each aircraft route selected, we determine its *earliest hub arrival* (EHA) time. This time is composite-dependent, as the route's interaction with other aircraft routes (within ramp transfer composites) might affect the EHA time.

First, we consider the case of an aircraft route in a *non-ramp transfer* composite, c. For a single-leg aircraft route (f_1, r_1) flying from gateway i to hub h, its EHA time is given by:

$$EHA_c(f_1, r_1) = EPT(i) + \beta_{ih}^{f_1},$$

where $EPT(i)$ is the Early Pickup Time for gateway i and $\beta_{ih}^{f_1}$ is the block time (i.e., taxi time plus flying time) for fleet type f_1 flying from i to h. For a double-leg aircraft route (f_2, r_2) flying from i through j to h, the EHA is:

$$EHA_c(f_2, r_2) = \max\left[EPT(j), EPT(i) + \beta_{ij}^{f_2} + \tau^{f_2}\right] + \beta_{jh}^{f_2},$$

where τ^{f_2} is the *turn time* for fleet type f_2.

Next, we determine the EHA for aircraft routes included in *ramp transfer* composites. Consider a single-leg aircraft route (f_3, r_3) flying from i to h involved in a ramp transfer at gateway i. Define the *Package Release Time*, $PRT_c(f_3, r_3, i)$, as 30 minutes following the latest arrival time of aircraft carrying packages to be transferred to (f_3, r_3). Thus, the EHA

for this aircraft route is:

$$EHA_c(f_3, r_3) = \max\left[EPT(i), PRT_c(f_3, r_3, i)\right] + \beta_{ih}^{f_3}.$$

For a double-leg aircraft route (f_4, r_4) flying from i through j to h and part of a ramp transfer at gateway j, the EHA is:

$$EHA_c(f_4, r_4) = \max\left[EPT(j),\ PRT_c(f_4, r_4, j),\ EPT(i) + \beta_{ij}^{f_4} + \tau^{f_4}\right] + \beta_{jh}^{f_4}.$$

Without explicitly including dynamic landing constraints, **ARM**'s solution yields an arrival pattern that satisfies each hub's target landing rate. Figure 4-9 shows the arrival pattern at the busiest hub (based on the EHAs of each aircraft route in the solution). We can freely shift any arriving aircraft route to a later time provided it does not arrive later than the hub's sort start time. Ramp transfers to this aircraft route are not affected by the shift as this simply increases the on-ground time for the aircraft at the ramp transfer gateway. For two-leg routes, the shift only affects the second-leg of the route so transfers from this aircraft to other aircraft routes are not affected.

The **ARM** solution's arrival pattern at the carrier's central hub is shown in Figure 4-9. The allowable arrival rate is 6.75 planes per 15 minute interval. East Coast planes must arrive by time 6.37 and West Coast planes must arrive by time 7.38. Two intervals violate the allowable arrival rate: $[5.25, 5.5]$ and $[6.25, 6.5]$. We may shift two of the aircraft routes from the $[5.25, 5.5]$ interval to the $[5.5, 5.75]$ interval. One of the aircraft routes in the $[6.25, 6.5]$ interval is a west coast route, and we may shift it to the $[6.5, 6.75]$ interval (i.e., after the non-west coast sort start time) and shift four aircraft routes from the $[5.75, 6]$ interval to the two successive intervals. The result is an arrival pattern that satisfies the maximum average landing rate of 6.75 airplanes per 15 minute interval.

Modeling Staggered Arrivals

While **ARM** finds valid arrival patterns on the current data set, we might not want to leave it to chance. We build dynamic constraints that account for aircraft arrivals at each hub, volume arrival at each hub, or both.

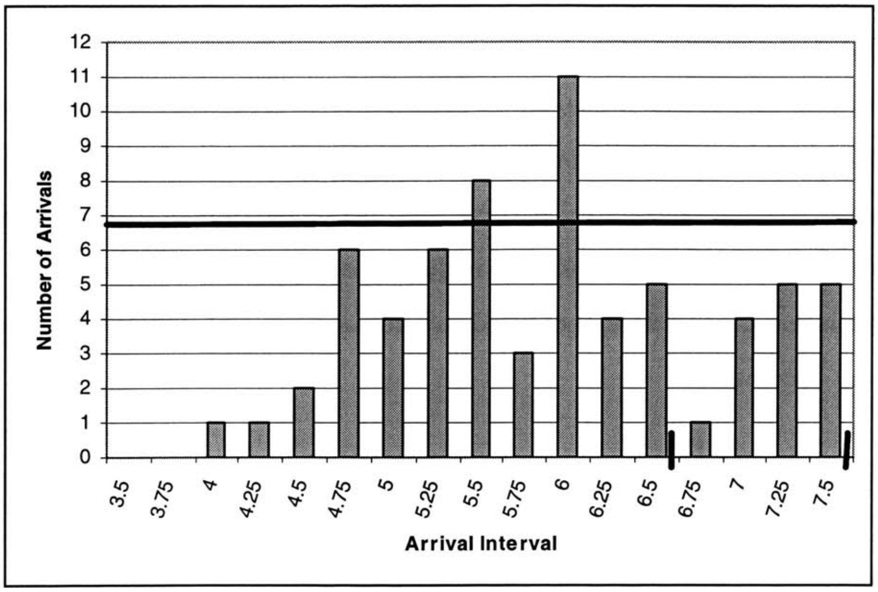

Figure 4-9: Arrival profile at central hub

For each hub, let T_h be the number of intervals of fixed duration during which aircraft may land. Define a_{ht} to be the allowable number of arrivals during the t^{th} interval at hub h. Define the indicator:

$$\lambda_c^{frt} = \begin{cases} 1 & \text{if } EHA_c(f,r) \text{ occurs during or after interval } t \\ 0 & \text{otherwise.} \end{cases}$$

The following cumulative arrival constraints create arrival limits that become more restrictive as we near the sort start time:

$$\sum_{c \in \mathcal{C}_P} \sum_{(f,r) \in c} \lambda_c^{frt} \gamma_c^{fr}(\underline{h}) v_c \leq \sum_{s=t}^{T} a_{hs} \qquad t = 1, ..., T,\ h \in H, \tag{4.1}$$

where $\mathcal{C}_P$ is the set of pickup composites; $\gamma_c^{fr}(\underline{h})$ is the number of times aircraft route (f,r), which lands at hub h, is included in composite c; v_c is the decision variable for the number of times we select composite c; and H is the set of hubs. In the final interval, (4.1) forces the number of arrivals not to exceed a_{hT}. The number of arrivals in the final two periods cannot exceed $a_{hT} + a_{h,T-1}$. While this allows excess arrivals in the $T - 1^{st}$ interval, these routes may be deferred by manually shifting flights to a later time. This will not affect the feasibility of the route, since the timing for pickups and ramp transfers will still be satisfied.

A related requirement is to ensure that the volume arriving to the hub is balanced over the period prior to sort start. While the staggered arrival constraints (4.1) ensure a steady landing rate, it might be the case that small planes arrive earlier and larger planes arrive later, resulting in an undesirable package arrival pattern that has more volume arriving closer to the sort start time. We modify constraints (4.1) to include the capacity assigned to each aircraft route and the target interval *capacities*, measured in packages (or containers) and denoted by $\widehat{a}_{ht}$. The constraints become:

$$\sum_{c \in C_P} \sum_{(f,r) \in c} \lambda_c^{frt} \gamma_c^{fr}(\underline{h}) u_r^f v_c \leq \sum_{s=t}^{T} \widehat{a}_{hs} \qquad t = 1, ..., T, \; h \in H. \tag{4.2}$$

When we build composites in practice, we know *a priori* the volume that may be carried by each aircraft route in a composite. Thus, we can replace $\gamma_c^{fr} u_r^f$ in (4.2) with $\widehat{u}_c^{fr}$, the total flow assigned *a priori* to fleet route (f, r) in composite c.

4.4 Scenario Analysis

The run times and robustness of **ARM** make it well-suited for analyzing the effects of changes to the data or to the overall operating strategy of the carrier. In this section, we examine two scenarios. The first uses **ARM** to assess the composition of the carrier's fleets. The second uses **ARM** to explore the effect of changing from a multi-hub network to a single-hub network.

4.4.1 Ideal Fleet Mix

In this scenario, we provide an unlimited number of aircraft of each fleet type and let **ARM** select what it determines to be the ideal mix of aircraft. As the carrier considers future aircraft requirements, we can also include fleet types not currently in the carrier's inventory. We initially consider the case of minimizing operating cost followed by the case of minimizing operating cost plus ownership cost. The solution is required to balance, but we remove the restriction for it to balance against the Second-Day Air network (which would otherwise force it to select aircraft used in the SDA network).

	Min Operating Cost	Min Operating + Ownership Cost
Operating Cost	5.28%	-18.89%
Aircraft Ownership Cost	-23.53%	23.95
Total Cost	-16.14%	9.36%
# Aircraft	2.26%	-17.29%

Table 4.10: Ideal Fleet Scenario: improvement over **ARM** with existing fleets

In Table 4.10, we show the ideal fleet mix chosen by the **ARM** solution relative to the **ARM** solution found with limited aircraft availability. By selecting the ideal fleet mix with minimum operating cost, **ARM** reduces operating cost by 5.28% and uses fewer aircraft and only three fleet types. These aircraft types, shown in Figure 4-8, have large ownership costs, which explains the poor performance of this solution with respect to ownership cost. When we include ownership cost in the objective, we see a shift in the fleet types that are selected (see Figure 4-10). In fact, of the 130 aircraft chosen when minimizing with respect only to operating cost, only 18 are selected when ownership cost is included in the objective function.

4.4.2 Single Hub Operations

Next, we consider the effect of consolidating all air operations through a single hub using the carrier's current inventory of aircraft. All gateway-hub demands are sent through this single hub. We enforce the same level-of-service requirements and hub timing requirements as described earlier. Once again, we enforce aircraft balance but do not impose the restriction of balancing against an existing SDA network.

Regardless of the objective, the single-hub strategy results in higher cost. Two things drive this higher cost. First, many gateway-to-gateway demands must be flown a longer distance because the "regional" hubs are no longer available. Second, demand either originating or terminating at a regional hub must now be handled on one additional route. With multiple hubs, demand originating at a gateway and terminating at a hub is handled once: on a pickup route. Once it reaches the hub, this volume is no longer flown in the NDA network. With a *single* hub, this demand is flown on a pickup route to the central hub and on a delivery route to the regional hub, requiring additional aircraft usage.

The result demonstrates two key points. First, it illustrates the effectiveness of regional hubs in reducing the travel time between many gateway pairs and in reducing the volume that

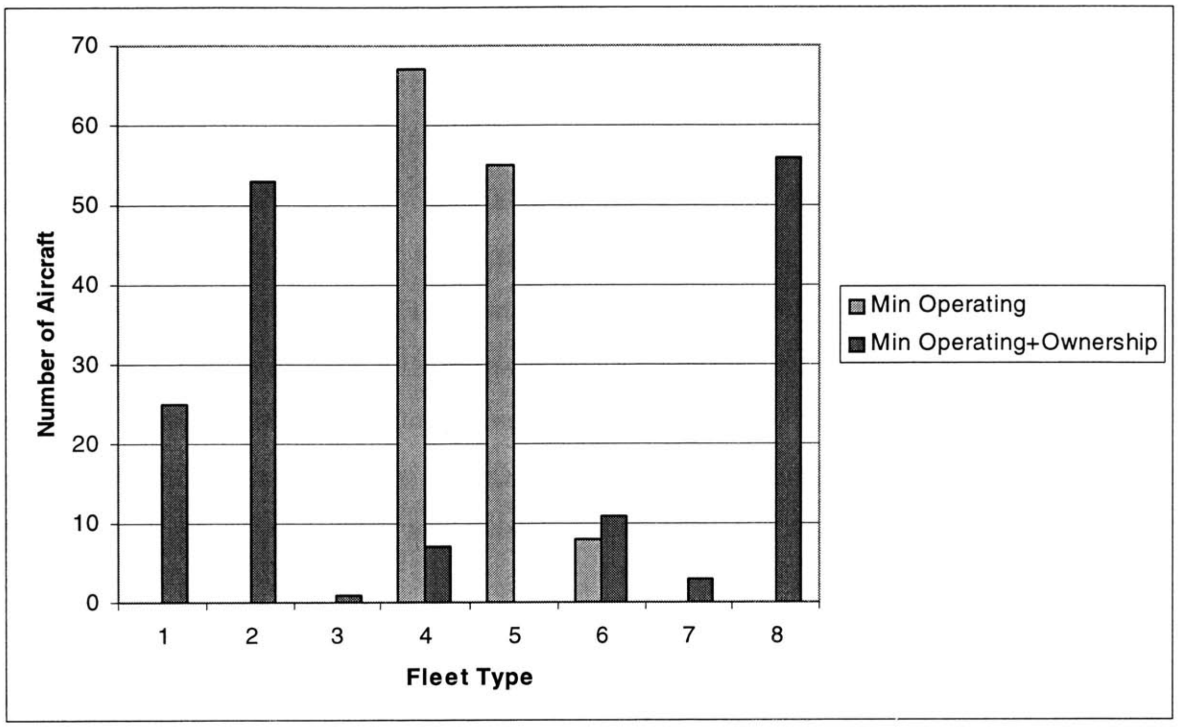

Figure 4-10: Aircraft usage for ideal fleet mix scenario

	Min Operating Cost	Min Operating + Ownership Cost
Operating Cost	-26.04%	-9.59%
Aircraft Ownership Cost	-8.95%	-32.14%
Total Cost	-13.33%	-24.46%
# Aircraft	-18.80	-15.04%

Table 4.11: Single Hub Scenario: improvement from **ARM** solution with multiple hubs

moves on each side (i.e., pickup or delivery) of the air network. Second, it demonstrates the potential for using the model to analyze the effect of adding additional hubs; expanding or contracting the capacity of existing hubs; or creating mini-hubs, which are gateways where packages may be transferred between planes (not sorted) and where routes may terminate. It does not suggest, however, that a single-hub operation has lower cost, as **ARM** only determines the cost of owning and operating aircraft. The cost increase must be considered with all relevant cost changes, such as the cost savings that result from operating fewer hubs.

4.5 Implementation at the United Parcel Service

During the past year, the Operations Research Group at UPS has implemented a route and schedule generation system with the Aircraft Routing Model as its core component. The initial version of the model contains many of the features contained in this and the previous chapters. Missing from the initial implementation are three- and four-route ramp transfer composites and drop-off composites, both of which promise to yield significant additional reductions in operating cost and number of aircraft used.

The results from initial validation efforts are very promising. In the planning of the peak season for Nov-Dec, 1999, **ARM** utilized seven fewer leased aircraft than the solution generated manually by the peak planners. With lease costs in the tens of thousand of dollars per plane block hour, the savings over the two month peak season would be considerable.

While the **ARM** solution was not actually used (in fact, it was generated *during* the peak season), the results were so promising that **ARM** is being used to support the generation of the 2000 peak season plan. In addition, the long-range planning group is using **ARM** to support the development of its plan for 2005. In the long-range planning context, saved aircraft are even more valuable as they represent assets the company does not have to purchase.

4.6 Summary

The Aircraft Routing Model (**ARM**) is an effective approach for developing aircraft routing plans for the largest express shipment networks. We have demonstrated that, for the United Parcel Service's domestic Next-Day Air network, **ARM** generates solutions with significantly lower operating cost and dramatically lower aircraft ownership cost. Moreover, it achieves this with fast run-times, making **ARM** an effective component of a decision support system for the network planners. The concept behind constructing composites is highly attractive for handling the complexity of practical problems. Composites, by their design, implicitly capture the difficult constraints of network design formulations and they can implicitly capture carrier-specific constraints simply by building necessary rules into the composite construction routines. While we have demonstrated the success of this formulation strategy on a carrier-specific instance of **ESSND**, we must next take steps to formalize the theory behind composite variable formulations and to generalize the approach for broader classes of network design and fixed charge problems.

Chapter 5

Strength of the Aircraft Routing Model

In Chapter 2, we presented the Express Shipment Service Network Design (**ESSND**) problem formulation, which contains decision variables for aircraft routes and decision variables for package flows. We highlighted earlier approaches for solving this problem, including decomposition strategies and heuristic solution approaches. In Chapter 3, we presented the idea of solving a "routes-only" model and presented a formulation whose variables were groupings of aircraft routes (i.e., *composite* variables). The practical impact of the new formulation, which we call the Aircraft Routing Model (**ARM**), was clearly demonstrated in Chapter 4.

In this chapter, we develop the theoretical link between **ESSND** and **ARM**. As shown in Figure 5-1, we transition from **ESSND** to **ARM** via an intermediate "routes-only" formulation that we call **RO**. The inclusion of this intermediate formulation serves two purposes. First, it provides a clear method for understanding the definition of composite variables. Second, it provides a convenient way to demonstrate the strength of **ARM** relative to **ESSND** by showing **RO** is at least as strong as **ESSND**; that **ARM** is at least as strong as **RO**; and that, in most cases, the strength is *strict.* In the integer programming sense, all three models are shown to be equivalent.

In addition to strength, other dimensions characterize this "family" of models. The size of the set of design variables *increases* as we move towards **ARM**, while the size of the package

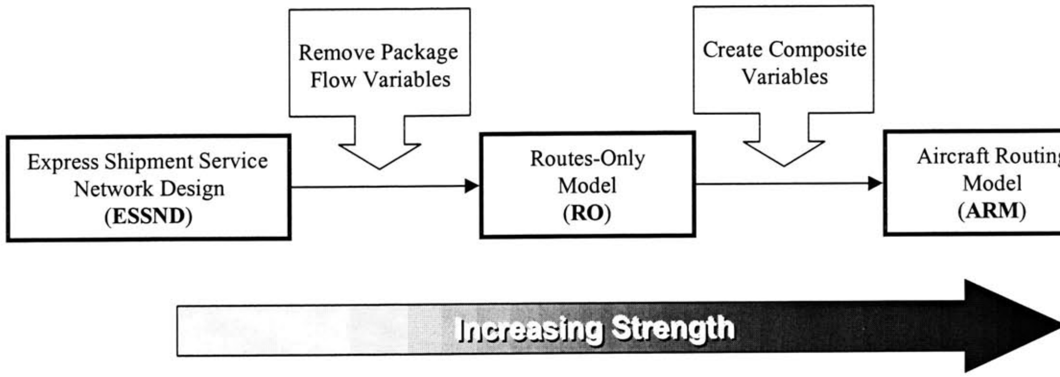

Figure 5-1: Transitioning from **ESSND** to **ARM** via an intermediate "routes only" model (**RO**)

flow variable set drops (to zero) for **RO** and **ARM**. The amount of information contained in the decision variables *increases* in **RO,** as the design variables capture information about package flows. The amount of information further increases in **ARM**, as the variables capture information about the interactions between aircraft routes and about which commodities are satisfied by the composite variable. With this increase in variable complexity comes a corresponding *increase* in the difficulty associated with generating those variables and including them in the model.

It is the complexity built into the composite variables that helps when using **ARM** for *realistic* instances of the problem. For example, among the operational factors we have encountered in this research is the need to work with *containers* of packages rather than individual packages. The underlying package flow problem must allow only *integral* solutions, as fractions of containers (each of which may hold hundreds or thousands of packages) cannot be absorbed into the solution the way fractions of packages can. **ARM** easily handles such requirements while **ESSND**'s already fragile tractability is destroyed.

Rules regarding ramp transfers include limits on where such transfers may occur, timing requirements for the transfer, and limits on the amount of volume that can be moved between planes. In **ARM**, satisfying these constraints (and any other practical constraints introduced by the planners) is simply a function of generating composite variables that satisfy the desired operational constraints with no structural changes required for the **ARM** formulation. Using the restricted set of columns in **ARM** generally improves the model's run time while building the same restrictions into the **ESSND** formulation makes it more difficult to solve.

Given this spectrum of models, the choice of which model to use depends on problem context. Factors affecting this choice include the size of the problem instance, running time requirements, number and detail of operational restrictions, flow variables that are integral or continuous, and the ease with which routes and composites are generated. In some cases, the explicit representation of flows and design variables might be the best approach (i.e., **ESSND**), in other cases, methods that capture flow variables implicitly in the design variables might be preferred. In the case of large-scale express shipment planning problems, combining as much of the complexity into the design decision variables proves to be the most effective strategy for quickly generating very good solutions.

The organization of this chapter follows the flow shown in Figure 5-1. We first re-state the **ESSND** formulation. We then characterize a method of representing aircraft routes that guarantees *feasible* flows. This characterization leads to the intermediate formulation, **RO**, which is equivalent to **ESSND** and has a stronger linear programming relaxation. Next, we examine the benefits of combining variables from the **RO** formulation to yield composite variables, which serve as the basis of the **ARM** formulation. We show the equivalence of **RO** and **ARM** and establish that the linear programming relaxation for the latter provides a bound not worse than that of **RO**. Finally, we show how the implementation of **ARM** described in Chapters 3 and 4 generates the optimal solution to the **ESSND** problem under an operational assumption enforced by planners. We refer to this model simply as *restricted* **ARM** because we restrict the composites that are included in the model according to the rules presented in Chapter 3.

5.1 ESSND Formulation

We begin by stating several assumptions enforced throughout the chapter:

- Package flow costs are zero
- All pickup routes end at a hub and all delivery routes begin at a hub

- Demands are specified by gateway-hub commodities
- No ramp transfers are permitted

The first three assumptions were enforced in earlier chapters. The fourth assumption allows us to construct the routes-only model, **RO**, and to use it to prove the equivalence of **ESSND** and **ARM**. Because the number of ramp transfer opportunities is limited (due to the timing of routes, the availability of excess capacity on aircraft, and the number of locations at which ramp transfers are possible), we temporarily set them aside, recognizing that each ramp transfer configuration can be represented as a composite variable and included in **ARM**.

In the formulations, proofs, and examples that follow, we distinguish between routes, aircraft routes, and paths. A *route* is an ordering of locations on the physical network. For instance a route is described by locations (i.e., either gateways or hubs) in the order in which they are visited. An *aircraft route* is the combination of a specific aircraft (fleet) type and a route. Not all fleet types can fly every route. Finally, a *path* is used to describe package flows through the underlying time-space network, the construction of which is described in Chapter 2 and Kim, et al. [52].

In Chapter 2, we presented **ESSND** by explicitly describing the coefficients for each constraint, and in Chapter 3, we described procedures for generating aircraft routes. In this chapter, for the sake of conciseness, we present the formulation using matrix notation. We use the following sets, matrices, data, and decision variables:

Sets

F	Set of fleet types
H	Set of hubs
G	Set of gateways
R^f	Set of routes flown by fleet type $f \in F$; the route set consists of disjoint sets for pickup routes and delivery routes, denoted by R^f_P and R^f_D
A	Set of flight arcs in the time-space network
$\mathcal{K}$	Set of commodities to flow through the network

Constraint Matrices

- **U** Mapping of aircraft route capacities u_r^f to arcs in A
- $\mathcal{N}^k$ Node-arc incidence matrix for commodity $k \in \mathcal{K}$
- **B** Balance constraint matrix
- **A** Landing (**A**rrival) constraint matrix
- **N** Plane count (**N**umber) constraint matrix

Indicator Variable

$$\delta_{ij}^{fr} = \begin{cases} 1 & \text{if aircraft route } (f,r) \text{ contains flight arc } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

Right-Hand-Side Data

- $\mathbf{b}^k$ Demand vector for commodity k with components b_i^k, $i \in N$
- **a** Vector of hub arrival capacities with components a_h, $h \in H$
- **n** Vector of fleet sizes with components n_f, $f \in F$

Decision Variables and Costs

- **y** Vector of aircraft route variables with components $y_r^f \in \mathbb{Z}_+$, $r \in R^f$, $f \in F$
- **x** Vector of package flow variables with components $x_{ij}^k \geq 0$, $(i,j) \in A$, $k \in \mathcal{K}$
- **d** Vector of aircraft route costs with components d_r^f, $r \in R^f$, $f \in F$

The **ESSND** formulation is:

$$\min \mathbf{d}'\mathbf{y} \tag{5.1}$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} x_{ij}^k - \sum_{f \in F} \sum_{r \in R^f} \delta_{ij}^{fr} u_r^f y_r^f \leq 0 \quad (i,j) \in A \tag{5.2}$$

$$\mathcal{N}^k \mathbf{x}^k = \mathbf{b}^k \quad k \in \mathcal{K} \tag{5.3}$$

$$\mathbf{By} = \mathbf{0} \tag{5.4}$$

$$\mathbf{Ay} \leq \mathbf{a} \tag{5.5}$$

$$\mathbf{Ny} \leq \mathbf{n} \tag{5.6}$$

$$y_r^f \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F \tag{5.7}$$

$$x_{ij}^k \geq 0 \quad (i,j) \in A,\ k \in \mathcal{K}. \tag{5.8}$$

The *forcing* constraints (5.2) limit the flow on each flight arc to the capacity provided by aircraft assigned to that flight arc. Constraints (5.3) ensure conservation of flow for all commodities. The aircraft balance constraints (5.4) force the number of planes of a given fleet type taking off from a location (gateway or a hub) to be offset by the same number of planes of that fleet type landing at that location. The *landing capacity* constraints (5.5) force the number of pickup routes landing at hub h to be less than a specified landing capacity, a_h. The *plane count* constraints (5.6) restrict the usage for each fleet type f to be less than the available number of aircraft of that fleet type, n^f. Constraints (5.7) and (5.8) enforce the *integrality* of the design variables and the *nonnegativity* of the flow variables, respectively. One set of **ESSND** constraints not included above are the *hub sort capacity* constraints, which we assume to be satisfied under the pre-assignment of gateway-hub demands.

Without ramp transfers, when packages are loaded onto an aircraft they remain on the aircraft until they reach their destination. We re-define the package flow variables accordingly: we simply assign demands to *routes* (versus arcs) and ensure that the fleet types assigned to fly the routes provide sufficient capacity to carry the demands. To reinforce the idea that commodity flows are identified by paths between gateways and hubs, we represent each *commodity* as a gateway-hub pair (g, h). The commodity set, $\mathcal{K}$, is split into two disjoint sets, $\mathcal{K}_P$ and $\mathcal{K}_D$, which correspond to the pickup side and delivery side, respectively. We define x_r^{gh} to be the amount of (g, h) demand assigned to route r. The set $R(g, h)$ includes all routes that connect gateway g with hub h ($R_P(g, h)$ and $R_D(g, h)$ for the pickup and delivery sides, respectively). With package flows now assigned based on path (route) flows, **ESSND** is rewritten as the following (**ESSND-R**):

$$\min \mathbf{d'y} \tag{5.9}$$

$$\text{subject to} \quad \sum_{(g,h)\in\mathcal{K}} x_r^{gh} - \sum_{f\in F} u_r^f y_r^f \leq 0 \quad r \in R \tag{5.10}$$

$$\sum_{r\in R_P(g,h)} x_r^{gh} = b_P^{gh} \quad (g,h) \in \mathcal{K}_P \tag{5.11}$$

$$\sum_{r \in R_D(g,h)} x_r^{gh} = b_D^{gh} \quad (g,h) \in \mathcal{K}_D \tag{5.12}$$

$$\mathbf{By} = \mathbf{0} \tag{5.13}$$

$$\mathbf{Ay} \leq \mathbf{a} \tag{5.14}$$

$$\mathbf{Ny} \leq \mathbf{n} \tag{5.15}$$

$$y_r^f \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F \tag{5.16}$$

$$x_r^{gh} \geq 0 \quad r \in R,\ (g,h) \in \mathcal{K}. \tag{5.17}$$

Constraints (5.10) are the *forcing* constraints. There is one forcing constraint for each route, compared to one forcing constraint for each flight arc in the original formulation, **ESSND**. Constraints (5.11) ensure that all the (g, h) pickup demands are fully assigned to pickup routes. Similarly, constraints (5.12), ensure all delivery demands are fully assigned to delivery routes. Constraints (5.13)-(5.15) are the *balance*, *landing*, and *plane count* constraints as described earlier.

There is one critical difference between these formulations. In **ESSND**, packages flow and aircraft are assigned to arcs in a time-space network. Depending on the speed of the fleet types, different fleet types flying the same route can be represented by distinct flight arcs. The timing of the aircraft routes is only important at intermediate gateway locations, where packages may be transferred from one plane to another (i.e., ramp transfers). When we remove ramp transfers, the timing of the flight legs at these intermediate locations is no longer important. We are simply interested in assigning commodities to routes that connect the commodity's origin with its destination.

Definition 15 *We say two formulations are* ***equivalent*** *if, for any feasible solution to one, there is a corresponding feasible solution to the other with equal cost (and vice versa).*

The direct consequence of this definition is that optimal solutions to equivalent formulations result in the same objective function value. The LP relaxations of equivalent (mixed) integer programming formulations may yield objective function values that are *not* the same, however. One criterion for choosing a formulation is the closeness of the linear programming relaxation to the convex hull of feasible integer solutions. We refer to this closeness as a formulation's *strength.* This is a relative measure used to compare formulations by establishing that one formulation is *stronger than* another. A stronger LP relaxation allows us to more quickly search for the optimal integer solution and to prove its optimality.

Definition 16 *Consider two equivalent (mixed) integer programming formulations A and B. Let A_{LP} and B_{LP} represent the LP relaxations of A and B. A_{LP} is said to be* ***at least as strong as*** *B_{LP} if we can map any feasible solution of A_{LP} to a feasible solution of B_{LP} with the same cost. If A_{LP} is at least as strong as B_{LP}, then A_{LP} is said to be* ***stronger than*** *B_{LP} if there exists a feasible solution to B_{LP} for which there is no corresponding feasible solution to A_{LP} with the same cost.*

We turn next to establishing the equivalence of **ESSND** and **ESSND-R**. The non-ramp transfer assumption implies that once packages are loaded on an airplane, they will remain on the airplane until they reach their destination. We can, therefore, examine the flows on arcs incident to the hub to establish the equivalence of the two formulations.

Lemma 17 *The mixed integer program* ***ESSND*** *and the integer program* ***ESSND-R*** *are equivalent.*

Proof. Any integral assignment of aircraft routes, $\widehat{\mathbf{y}}$, that satisfies (5.4)-(5.6) and (5.7) in **ESSND** also satisfies (5.13)-(5.15) and (5.16) in **ESSND-R**. The cost of the solution $\widehat{\mathbf{y}}$ is the same in both formulations, since aircraft route costs are identical. So in order to establish equivalence, we must show that any solution $(\widehat{\mathbf{x}}, \widehat{\mathbf{y}})$ to **ESSND** has a corresponding set of feasible package flows in **ESSND-R** using the same set of aircraft routes, $\widehat{\mathbf{y}}$, and vice versa.

For a fixed set of aircraft routes, $\widehat{\mathbf{y}}$, assume the arc flows in the time-space network satisfy the **ESSND** forcing constraints. All commodities must flow on some arc incident to the hub. Consider the set of arcs corresponding to a given route, r^*, with one arc for each aircraft type that can fly the route. Since each flight arc satisfies forcing constraint (5.2), summing the constraints over all flight arcs corresponding to route r^* yields the **ESSND-R** forcing constraint (5.10) for route r^*. Finally, by constraint (5.3), the **ESSND** solution ensures that the demand for each commodity is flown from its origin to destination. The entire demand for each commodity must flow on some arc(s) incident to the hub. Thus, summing the arc flows for all aircraft routes (f, r) that connect a given gateway-hub pair must account for all the demand, b_P^{gh} (for the pickup side). Therefore, **ESSND-R** constraints (5.3) are satisfied for each commodity (g, h).

Conversely, any solution to **ESSND-R** has package flows aggregated by route, which can be *disaggregated* by fleet type such that the resulting flows satisfy the capacities of those aircraft routes. We then map these path flows to the arcs in the **ESSND** time-space network and the forcing constraints (5.2) are satisfied for each arc in the time-space network. For each commodity, this mapping assigns flows for commodity k (i.e., (g, h)) to paths through the time-space network. The total flow both from the origin and to the destination is b^k (for pickup routes) and conservation of flow is maintained at all intermediate points on those paths. Hence, the **ESSND** flow balance constraints (5.13) are satisfied. ■

By relaxing the integrality requirement, the previous proof leads directly to the following:

Corollary 18 *The linear programming relaxations of* ***ESSND*** *and* ***ESSND-R*** *are equivalent.*

This simplified formulation, **ESSND-R**, exploits the assumption that no ramp transfers are allowed, providing a starting point from which to create a model consisting only of aircraft route variables.

5.2 The Routes-Only Model

Due to our assumptions about ramp transfer, a given aircraft route will carry only the commodities corresponding to the gateways it visits and the hub at which it terminates (for pickup routes) or originates (for delivery routes). The problem, then, is to identify the *available capacity* an aircraft route may use to carry some demands. For double-leg routes, there may be an infinite number of ways to divide the available capacity between the two gateway-hub commodity demands. For single-leg routes, the entire capacity of the aircraft route is available to move the demand between the gateway and hub.

5.2.1 Extreme Routes

We associate with each aircraft route a set of *extreme routes*, each of which specifies an *extreme* allocation of the aircraft's capacity along that route. Extreme routes are expressed in terms of the capacity made available to the commodities; the actual flow might be less than the available capacity. In the case of double-leg routes, at one extreme we give preference to the route's first location, loading the aircraft with as much of the demand from that gateway as possible and using the excess capacity (if any) for the demand at the route's second location. At the other extreme, we give preference to the second location. Once again, we are specifying an extreme allocation of *capacity* and we are not specifying the actual flows.

To formalize the definition, consider a pickup aircraft route (f, r). The *available capacities* for the first extreme route corresponding to (f, r) are as follows:

$$\begin{aligned}\widehat{u}_{ih}^{1} &= \min\{b_P^{ih}, u_r^f\}\\ \widehat{u}_{jh}^{1} &= \min\{u_r^f - \widehat{u}_{ih}^{1}, b_P^{jh}\}.\end{aligned}$$

The available capacities for the second extreme route are:

$$\begin{aligned}\widehat{u}_{jh}^{2} &= \min\{b_P^{jh}, u_r^f\}\\ \widehat{u}_{ih}^{2} &= \min\{u_r^f - \widehat{u}_{jh}^{2}, b_P^{ih}\}.\end{aligned}$$

By taking a convex combination of a pair of extreme routes, we construct any possible allocation of an aircraft route's capacity to the two gateway-hub demands it can serve (proven in Lemma 20 below).

It is possible that the available capacities are the same for both extreme routes. This occurs when the total capacity of the route exceeds the *total* demand to be moved, that is $u_r^f \geq b_P^{ih} + b_P^{jh}$. Then the first nonzero element of each capacity vector is b_P^{ih} and the second nonzero element of each vector is b_P^{jh}.

In the case of single-leg routes, capacity is allocated to a single gateway-hub commodity, (i, h), and we have only a single extreme route. The available capacity for this extreme route is defined as:

$$\widehat{u}_{ih}^{1} = \min\{b_P^{ih}, u_r^f\}. \tag{5.18}$$

When the aircraft route capacity exceeds the total demand that can move on the aircraft

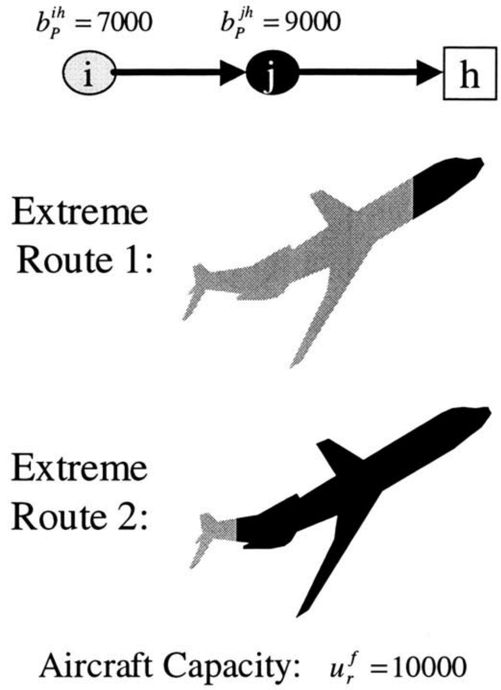

Figure 5-2: Available capacities for extreme routes corresponding to double-leg aircraft route

route (for single-leg routes and double-leg routes alike), the extreme route is identical to the original aircraft route with the capacity coefficient reduced. The value to which it is reduced equals the total demand that can be flown on that aircraft route. This act of lowering the capacity-demand coefficient is identical to what is commonly known as *coefficient reduction* (see Nemhauser et al. [63], for example).

Example 19 *Consider the aircraft route shown in Figure 5-2. The plane has a 10000 package capacity. The first gateway-hub commodity has a volume of 7000 packages and the second gateway-hub commodity has a volume of 9000 packages. The first extreme route picks up as much of the (i, h) demand as possible and uses the remaining capacity for the (j, h) demand. The available capacities associated with extreme route 1 are $\widehat{u}_{ih}^1 = 7000$ and $\widehat{u}_{jh}^1 = 3000$. The second extreme route gives preference to the (j, h) demand. The available capacities associated with extreme route 2 are $\widehat{u}_{ih}^2 = 1000$ and $\widehat{u}_{jh}^2 = 9000$.*

Using the same set of extreme routes, any package flow can be flown using the available

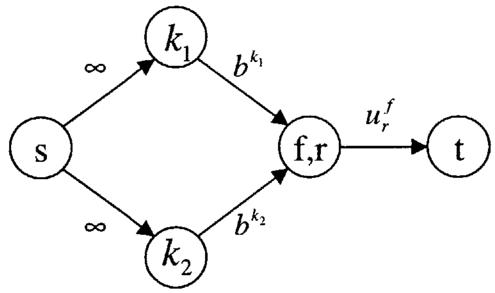

Figure 5-3: Maximum flow network for extreme routes

capacity created via a convex combination of the extreme routes. For instance, if the aircraft picks up a partial load, say 5000 packages from each location, we weight the first extreme point by $\frac{2}{3}$ and the second extreme point by $\frac{1}{3}$ to yield an available capacity of 5000 packages for each gateway-hub commodity. That is,

$$\frac{2}{3}\begin{pmatrix} 7000 \\ 3000 \end{pmatrix} + \frac{1}{3}\begin{pmatrix} 1000 \\ 9000 \end{pmatrix} = \begin{pmatrix} 5000 \\ 5000 \end{pmatrix}$$

Finally, using the same extreme routes, consider the case when the flow on the aircraft route doesn't use the entire capacity of the aircraft. Let the aircraft carry 2000 packages from each location. Then there is a range of multipliers that provide adequate capacity: $\lambda_1 \in \left[\frac{1}{6}, 1\right]$ *and* $\lambda_2 = 1 - \lambda_1$.

Extreme routes are aptly named as they are extreme point solution to a simple maximum flow problem. For a given double-leg aircraft route (f, r), we construct the network shown in Figure 5-3. We have one node corresponding to each commodity, k_1 and k_2; one node for the aircraft route, (f, r); a source node, s; and a sink node, t. We connect the source node to each commodity node with arcs having infinite capacity. We connect each commodity node to the aircraft route node with arcs having capacity equal to the commodity's demand, b^k. Finally, we connect the aircraft route node to the sink node with an arc having capacity u_r^f.

We augment the network with an uncapacitated arc from t to s, which yields a feasible flow graph. We are particularly interested in characterizing the set of extreme point solutions corresponding to the maximum flow. The maximum flow can be found by assigning a unit cost of -1 on the arc from t to s and zero cost to all other arcs in the network and solving as a minimum cost network flow problem. Extreme points of this problem are equivalently spanning trees on this graph (see Ahuja et al. [2]).

When flow in the network is positive, we immediately identify the arc from t to s as basic. When this positive flow equals u_r^f, we consider the case when the flow out of s uses both arcs and the case when it uses a single arc. Assume that the flow out of s uses *both* arcs. Then we have identified the second and third basic arcs, namely the two uncapacitated arcs from s. Among the remaining three arcs (the two from the commodity nodes and the one from the aircraft route node) we must identify at most one basic arc. At *most* one arc out of the commodity nodes can be *less* than its capacity, otherwise we would violate the spanning tree structure of the extreme point solution. This means that if commodity k_1 is picked up in full, the second commodity is flown in the amount $u_r^f - b^{k_1}$, and vice versa.

Assume now that the flow uses only one arc from s. Assume (w.l.o.g.) that the flow corresponds to commodity k_1. Then we have identified two basic arcs, one from t to s and the other from s to k_1. If $b^{k_1} > u_r^f$, then the arc from k_1 to (f, r) must be basic. The fourth basic arc is selected from the remaining arcs (not including (f, r) to t), both of which must be at their lower bounds. If $b^{k_1} < u_r^f$, then there must be additional flow of the other commodity to yield the total flow of u_r^f, which violates our assumption.

Thus, there are two extreme points corresponding to the maximum flow of u_r^f. The first corresponds to picking up as much of k_1 as possible and using the remaining (if any) capacity for commodity k_2. The second corresponds to picking up as much of k_2 as possible and using the remaining (if any) capacity for commodity k_1. In other words, these extreme points for the maximum flow are simply the extreme routes for the case when $b^{k_1} + b^{k_2} > u_r^f$.

Next, consider the case when the flow in the network is less than u_r^f. Then we immediately identify as basic both the arc from (f, r) to t and the arc from t to s. The only choice for the final two basic arcs requires the two capacitated arcs to be at their bounds, making arcs (s, k_1) and (s, k_2) basic. There are four choices for setting the capacitated arcs to their bounds: $(0, 0)$, $(b^{k_1}, 0)$, $(0, b^{k_2})$, and (b^{k_1}, b^{k_2}). The fourth corresponds to the maximum flow. Again, this single maximum flow extreme point is the same as the single extreme route for the case when $b^{k_1} + b^{k_2} \leq u_r^f$.

For routes consisting of more than two legs, these results generalize. We simply add additional nodes to the feasible flow graph and add arcs with the appropriate capacities. The extreme points corresponding to the maximum flow will define the extreme routes.

We are concerned only with the maximum flow extreme points because we are going to use extreme routes to specify *available capacity,* not actual flows. As such, we build the available capacities based on the maximum flow. The following result relates extreme routes to actual flows on aircraft routes and is central to the development of the routes-only formulation:

Lemma 20 *A package flow is feasible on an aircraft route (f,r) with capacity u_r^f* ***if and only if*** *it is feasible on some convex combination of the extreme routes of (f,r).*

Proof. (All arguments are presented in terms of *pickup* routes and the results apply similarly to delivery routes.) We consider three cases. The first is for single-leg routes, the second is for a double-leg route with capacity that exceeds its gateway-hub demands, and the third is for a double-leg route with gateway-hub demands that exceed its capacity. For each case, we show that for a given flow on an aircraft route, there is a convex combination of extreme routes on which that flow is also feasible, and vice versa.

Consider the **first case** when (f,r) is a single-leg route from i to h with flow $\widehat{x}_{ih}^{fr}$. We have a single extreme route with $\widehat{u}_{ih}^1 = \min\{b_P^{ih}, u_r^f\}$ and the total flow on (f,r) cannot exceed demand b_P^{ih}. Any flow less than u_r^f cannot be greater than $\widehat{u}_{ih}^1$ and vice versa.

Consider the **second case** when (f,r) is a double-leg route from i to j to h with $b_P^{ih} + b_P^{jh} \leq u_r^f$. There is a single extreme route with capacities $\widehat{u}_{ih}^1 = b_P^{ih}$ and $\widehat{u}_{jh}^1 = b_P^{jh}$. Given any feasible flow $\left(\widehat{x}_{ih}^{fr}, \widehat{x}_{jh}^{fr}\right)$ on this double-leg route, we have $\widehat{x}_{ih}^{fr} \leq b_P^{ih} = \widehat{u}_{ih}^1$ and $\widehat{x}_{jh}^{fr} \leq b_P^{jh} = \widehat{u}_{jh}^1$ and the flow is feasible with respect to the extreme route. Given a flow $\left(\overline{x}_{ih}^{fr}, \overline{x}_{jh}^{fr}\right)$ that is feasible with respect to the extreme route we have $\overline{x}_{ih}^{fr} \leq \widehat{u}_{ih}^1$ and $\overline{x}_{jh}^{fr} \leq \widehat{u}_{jh}^1$. Summing, we get $\overline{x}_{ih}^{fr} + \overline{x}_{jh}^{fr} \leq \widehat{u}_{ih}^1 + \widehat{u}_{jh}^1 \leq u_r^f$ and the flow is feasible with respect to the aircraft route, (f,r).

The **third case** is when (f,r) is a double-leg route from i to j to h when $b_P^{ih} + b_P^{jh} > u_r^f$. Given a feasible flow on (f,r), we have $\widehat{x}_{fr}^{ih} + \widehat{x}_{fr}^{jh} \leq u_r^f$. The flow of commodity (i,h) satisfies both $\widehat{x}_{ih}^{fr} \leq u_r^f$ and $\widehat{x}_{ih}^{fr} \leq b^{ih}$ and it follows that $\widehat{x}_{ih}^{fr} \leq \widehat{u}_{ih}^1$. If $\widehat{u}_{ih}^2 < \widehat{x}_{ih}^{fr} \leq \widehat{u}_{ih}^1$, we can find λ_1

and λ_2 such that

$$\begin{aligned} \lambda_1 + \lambda_2 &= 1 \\ \widehat{u}^1_{ih}\lambda_1 + \widehat{u}^2_{ih}\lambda_2 &= \widehat{x}^{fr}_{ih}. \end{aligned} \tag{5.19}$$

These multipliers also provide sufficient capacity to cover the demand from j to h:

$$\begin{aligned} \widehat{x}^{fr}_{jh} &\leq u^f_r - \widehat{x}^{fr}_{ih} \\ &= u^f_r - \left(\widehat{u}^1_{ih}\lambda_1 + \widehat{u}^2_{ih}\lambda_2\right) \\ &= \lambda_1\left(u^f_r - \widehat{u}^1_{ih}\right) + \lambda_2\left(u^f_r - \widehat{u}^2_{ih}\right) \\ &= \lambda_1\widehat{u}^1_{jh} + \lambda_2\widehat{u}^2_{jh}. \end{aligned}$$

When $\widehat{x}^{fr}_{ih} \leq \widehat{u}^2_{ih}$, the second extreme route covers both demands and we let $\lambda_1 = 0$ and $\lambda_2 = 1$.

Conversely (for the third case), assume we have a flow, $(\overline{x}^{fr}_{ih}, \overline{x}^{fr}_{jh})$, that is feasible with respect to a convex combination of its extreme routes. That is, $\overline{x}^{fr}_{ih} \leq \widehat{u}^1_{ih}\lambda_1 + \widehat{u}^2_{ih}\lambda_2$ and $\overline{x}^{fr}_{jh} \leq \widehat{u}^1_{jh}\lambda_1 + \widehat{u}^2_{jh}\lambda_2$. Summing the two inequalities, we obtain

$$\begin{aligned} \overline{x}^{fr}_{ih} + \overline{x}^{fr}_{jh} &\leq \left(\widehat{u}^1_{ih} + \widehat{u}^1_{jh}\right)\lambda_1 + \left(\widehat{u}^2_{ih} + \widehat{u}^2_{jh}\right)\lambda_2 \\ &= u^f_r(\lambda_1 + \lambda_2) = u^f_r, \end{aligned}$$

which is the desired result. ■

To visualize the relationship between available capacity and package flows, we illustrate the third case of the proof (i.e., $b^{ih}_P + b^{jh}_P > u^f_r$) in Figure 5-4. The corner points marked with circles correspond to the extreme routes. The point $\left(x^1_{ih}, x^1_{jh}\right)$ requires a convex combination of the two extreme routes to allocate enough capacity for both commodity flows. The capacity provided by extreme route 2 provides adequate capacity for the commodity flows of point $\left(x^2_{ih}, x^2_{jh}\right)$. Thus, for any feasible flow on aircraft route (f, r), we can find some convex combination of extreme routes that allocates sufficient capacity for the specified commodity flows.

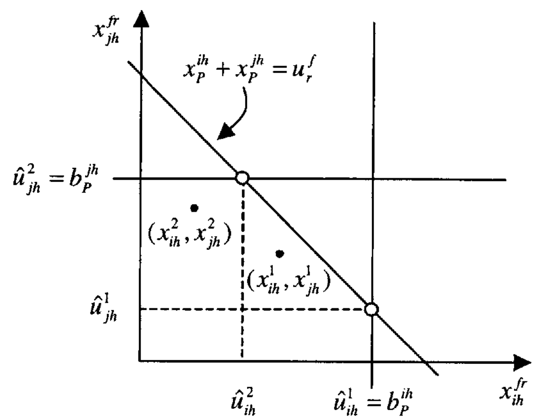

Figure 5-4: The relationship between extreme routes and feasible flows

5.2.2 RO Formulation

We introduce additional notation to create a formulation based on extreme routes. We define E to be the set of extreme routes as constructed above and we let w_e, $e \in E$, be the decision variables corresponding to the selection of each extreme route. The set E consists of two disjoint sets, E_P and E_D, corresponding to the pickup and delivery sides, respectively. We let δ_e^{fr} indicate which extreme routes correspond to aircraft route (f, r), setting it to 1 for extreme routes that correspond the aircraft route and 0 otherwise. For any aircraft route, the number of indicators with nonzero value is at most 2.

In this formulation, we require each aircraft route, built from its extreme routes, to be integral. That is, $\sum_{e \in E} \delta_e^{fr} w_e$ must yield an integer. The decision variables, w_e, are *not* necessarily integral. The number of decision variables in the new formulation is $|E|$, which is at most twice the number of aircraft route decision variables in **ESSND-R**.

The formulation for the routes-only model (**RO**) is given by:

$$\min \widehat{\mathbf{d}}'\mathbf{w} \tag{5.20}$$

$$\text{subject to} \quad \sum_{e \in E_P} \widehat{u}_{gh}^e w_e \geq b_P^{gh} \quad (g, h) \in \mathcal{K}_P \tag{5.21}$$

$$\sum_{e \in E_D} \widehat{u}^e_{gh} w_e \geq b_D^{gh} \quad (g,h) \in \mathcal{K}_D \tag{5.22}$$

$$\widehat{\mathbf{B}}\mathbf{w} = \mathbf{0} \tag{5.23}$$

$$\widehat{\mathbf{A}}\mathbf{w} \leq \mathbf{a} \tag{5.24}$$

$$\widehat{\mathbf{N}}\mathbf{w} \leq \mathbf{n} \tag{5.25}$$

$$\sum_{e \in E} \delta_e^{fr} w_e \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F. \tag{5.26}$$

Constraints (5.21) ensure that the total pickup capacity made available for pickup commodity (g, h) exceeds the demand for that commodity (referred to as the *pickup capacity-demand* constraints). Similarly, constraints (5.22) ensure that the total capacity made available for delivery commodity (g, h) exceeds the demand for that commodity (referred to as the *delivery capacity-demand* constraints). Constraints (5.23)-(5.25) are the *balance*, *landing*, and *plane count* constraints described earlier. Finally, constraints (5.26) ensure that each aircraft route constructed from its extreme routes is selected in integer multiples. Note that the decision variables w_e need not be integral, only the aircraft routes are integral.

5.2.3 Solution and Bounds

We compare feasible (optimal) solutions of **RO** with those of **ESSND-R**. The sets of feasible integral solutions are shown to be equivalent while the solution to the LP relaxation of **RO** gives a bound no worse than that of **ESSND-R**. Before stating the formal proofs, we introduce some preliminary observations that are used in the proofs.

An **RO** solution is mapped to a set of aircraft routes in **ESSND-R** as follows. Given a solution to **RO**, we construct a set of aircraft routes via the mapping

$$y_r^f = \sum_{e \in E} \delta_e^{fr} w_e. \tag{5.27}$$

Next, let $(\widehat{\mathbf{x}}, \widehat{\mathbf{y}})$ be a solution to **ESSND-R**. By Lemma 20, there exists a convex combination of the extreme routes of each aircraft route to cover the flow specified by $\widehat{\mathbf{x}}$. We let $\widehat{w}_e$ be the weight of extreme route e in such a convex combination. Then, by definition, (5.27) holds.

Consider any column of **RO**, which corresponds to some extreme route. The coefficients in this column corresponding to cost (5.20), aircraft balance (5.23), landing (5.24), and plane count (5.25) are *the same* as entries for the corresponding aircraft route column in **ESSND-R**. The differences in the columns are the coefficients for the capacity-demand constraints for pickup (5.21) and delivery (5.22) routes.

The mapping (5.27) preserves the feasibility of constraints involving only aircraft route variables. We illustrate this through the landing constraints. Let $\mathbf{A}_{(f,r)}$ be the column of the landing constraint matrix (5.14) in **ESSND** corresponding to aircraft route (f, r) and let $\widehat{\mathbf{A}}_e$ be the column of the landing constraint matrix (5.24) in **RO** corresponding to extreme route e. Then by the mapping in (5.27) we have

$$\widehat{\mathbf{A}}\mathbf{w} = \sum_{e \in E} \widehat{\mathbf{A}}_e w_e = \sum_{e \in E} \mathbf{A}_{(f,r)} \delta_e^{fr} w_e = \sum_{(f,r)} \mathbf{A}_{(f,r)} y_r^f = \mathbf{A}\mathbf{y}.$$

Given a solution to either **RO** or **ESSND**, if it satisfies the landing constraint in one formulation, it will satisfy the landing constraint in the other formulation. Similarly, this relationship holds for both the balance constraints, the plane count constraints, and the cost coefficients.

In addition, this mapping (5.27) preserves the integrality of aircraft routes. Given a solution $(\widehat{\mathbf{x}}, \widehat{\mathbf{y}})$ to **ESSND-R**, assume (f, r) has been selected n times; that is, $\widehat{y}_r^f = n$. Call the extreme routes corresponding to this aircraft route e_1 and e_2. For each of the n aircraft routes, Lemma 20 guarantees the existence of a convex combination the extreme routes of (f, r) that creates sufficient capacity to cover the flow assigned to that single aircraft route. For each of the n aircraft routes, we have such a convex combination of e_1 and e_2. If we let $\widehat{w}_{e_1}$ (and $\widehat{w}_{e_2}$) equal the sum of the n weights assigned to extreme route e_1 (and e_2), then $\widehat{w}_{e_1} + \widehat{w}_{e_2} = n$ and is, therefore, integral. Conversely, the integrality of an aircraft route constructed by the mapping (5.27) follows directly from the integrality of $\sum_{e \in E} \delta_e^{fr} w_e$ in the feasible **RO** solution.

We use these observations to establish the following relationship between the mixed integer program **ESSND-R** and the integer program **RO**:

Lemma 21 *The **RO** and **ESSND-R** are equivalent (mixed) integer programming formulations.*

Proof. Establishing equivalence requires proving that the mapping (5.27) maintains the feasibility of package flows. Assume we are given an **ESSND-R** solution $(\widehat{\mathbf{x}}, \widehat{\mathbf{y}})$. All package flows are assigned to aircraft routes. For *each* aircraft route, there exist a convex combination of extreme routes that provides available capacity to cover the flow assigned to that aircraft route (c.f. Lemma 20). Then for each gateway-hub (pickup) demand, summing the available capacities $\sum_{e \in E_P} \widehat{u}^e_{gh} w_e$ exceeds the total flow, which by (5.11) is b^{gh}_P (similarly for the delivery side). Thus, any feasible integer solution to **ESSND-R** has a corresponding integer solution in **RO** with the same cost.

Conversely, assume we are given a feasible **RO** solution, $\overline{\mathbf{w}}$, and we construct a feasible aircraft route solution to **ESSND-R** via (5.27). Lemma 20 establishes that a feasible flow on a convex combination of the extreme routes implies a feasible flow on each aircraft route. A feasible flow, therefore, exists on the set of routes specified by $\overline{\mathbf{y}}$. The forcing constraints (5.10) and the demand constraints (5.11) and (5.12) are, therefore, satisfied. The mapping yields an **ESSND-R** solution with the same cost as the **RO** solution. ■

The same mapping can be applied from a feasible solution of the **RO** LP relaxation to a feasible solution of the **ESSND-R** LP relaxation with the same cost. Using the same arguments as in the proof of the converse of Lemma 21 yields the following result:

Lemma 22 *The linear programming relaxation of **RO** is at least as strong as that of **ESSND-R**.*

There are cases when this strength is *strict* and the bound provided by the **RO** LP relaxation is strictly greater (tighter) than the bound provided by the **ESSND-R** LP relaxation. In Section 5.4, we present an example of such a case. In realistic instances of the problem, the conditions which cause strict improvement in the bounds are almost always satisfied.

5.3 Composite Variable Model

We have developed a formulation that explicitly models aircraft routes and ensures feasible package flows by providing sufficient capacity through weighted combinations of extreme routes. We take this one step further by combining routes into *composite variables*, each of which has sufficient capacity to carry some set of commodities. The combined routes might have excess capacity in the same way that **ESSND-R** allocated excess capacity. It is likely that further strengthening can occur by reducing coefficients in the composite variables for which excess

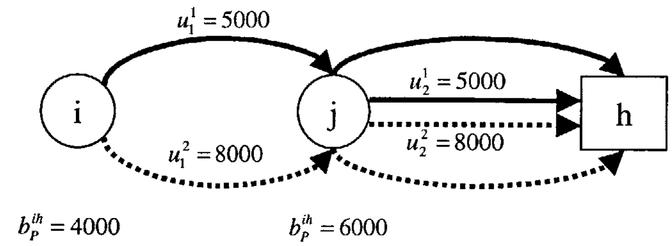

Figure 5-5: Two-gateway, one-hub network for the Composite Variable Example

capacity exists. We motivate this reformulation strategy via the mechanics demonstrated in the following example.

Example 23 (Composite Variable Example) *Consider the network shown in Figure 5-5. We must satisfy two gateway-hub demands. The first has a volume of* $b_P^{ih} = 4000$ *packages and the second has* $b_P^{ih} = 6000$*. We have one double-leg route from* i *to* j *to* h *and one single-leg route from* j *to* h*. We have two aircraft types, one with low capacity (5000 packages) and one with high capacity (8000 packages).*

For each of the double-leg routes we have two extreme routes (since their capacity does not exceed the sum of the gateway-hub demands). For each of the single-leg routes, we have one extreme route. The capacity demand constraints are given by:

$$\left[\begin{array}{cc|cc|c|c} 4000 & 0 & 4000 & 2000 & 0 & 0 \\ 1000 & 5000 & 4000 & 6000 & 5000 & 6000 \end{array}\right] \mathbf{w} \geq \left[\begin{array}{c} 4000 \\ 6000 \end{array}\right]$$

where the first row corresponds to commodity (i, h) *and the second corresponds to commodity* (j, h).

Dividing by the rhs yields capacity-demand coefficients of:

$$\left[\begin{array}{cc|cc|c|c} 1 & 0 & 1 & \frac{1}{2} & 0 & 0 \\ \frac{1}{6} & \frac{5}{6} & \frac{2}{3} & 1 & \frac{5}{6} & 1 \end{array}\right] \mathbf{w} \geq \left[\begin{array}{c} 1 \\ 1 \end{array}\right].$$

Columns 2 and 5 are identical with respect to capacity-demand coefficients. The difference between these columns is in the coefficients for the constraints not shown, specifically the aircraft *balance constraints. Because the two routes originate at different gateways, their aircraft balance coefficients will differ.*

For extreme routes corresponding to the **same** *aircraft route (columns 1 and 2, for example), the entries for constraints other than the capacity-demand constraints are identical. The only difference in the two extreme routes is how they allocate their capacity.*

By adding columns 3 and 5, we obtain a column with capacity-demand coefficients:

$$\begin{bmatrix} 1 \\ \frac{9}{6} \end{bmatrix}.$$

This indicates that by selecting the composite consisting of routes y_1^2 *and* y_2^1 *we have the available capacity to cover the entire demand of both commodities. Finally, because the* **rhs** *is 1, we can reduce the second coefficient from* $\frac{9}{6}$ *to 1.*

By adding this column to the existing set of decision variables we do not change the optimal integer solution. Any feasible solution (including the optimal solution) to the formulation without composites is still feasible (optimal). The impact comes when we consider the formulation consisting only of composite variables.

5.3.1 Formulation

To derive the composite variable formulation, we re-state the definition of composites and covers from Chapter 3:

Definition 24 *A* ***composite****, denoted by* c*, is a combination of distinct aircraft routes* (f, r), $f \in F$, $r \in R^f$. *Associated with* c *are the parameters* γ_c^{fr}, *which indicate the (integral) number of planes of fleet type* f *that fly route* r *in composite* c.

Definition 25 *A composite,* c*, is said to* ***cover*** $\mathcal{K}_c \subset \mathcal{K}$ *if there exists a feasible flow in* c *for the gateway-hub demands contained in* $\mathcal{K}_c$.

Let $\mathcal{C}$ be the set of all composite covers, which is likely to be too large to fully enumerate. The separation of aircraft routes into pickup and delivery routes allows us to divide the set of composites into two sets, $\mathcal{C}_P$ and $\mathcal{C}_D$, for the pickup side and delivery side, respectively. For any

$c \in \mathcal{C}$, we let $\delta_c^{gh} = 1$ for each commodity (g,h) covered by composite c (i.e., $(g,h) \in \mathcal{K}_c$). Each column of $\overline{\mathbf{B}}$ is constructed from the columns of $\mathbf{B}$ corresponding to the aircraft routes contained in c. If $\mathbf{B}_{(f,r)}$ is the column of $\mathbf{B}$ corresponding to aircraft route (f,r), then $\overline{\mathbf{B}}_c = \sum_{(f,r)} \gamma_c^{fr} \mathbf{B}_{(f,r)}$ and the elements of $\overline{\mathbf{B}}$ are integral ($\overline{\mathbf{A}}$ and $\overline{\mathbf{N}}$ are defined similarly). The composite variable formulation, which we call the Aircraft Routing Model (**ARM**), is defined as follows:

$$\min \overline{\mathbf{d}}'\mathbf{v} \tag{5.28}$$

$$\text{subject to} \quad \sum_{c \in \mathcal{C}_P} \delta_c^{gh} v_c \geq 1 \quad (g,h) \in \mathcal{K}_P \tag{5.29}$$

$$\sum_{c \in \mathcal{C}_D} \delta_c^{gh} v_c \geq 1 \quad (g,h) \in \mathcal{K}_D \tag{5.30}$$

$$\overline{\mathbf{B}}\mathbf{v} = \mathbf{0} \tag{5.31}$$

$$\overline{\mathbf{A}}\mathbf{v} \leq \mathbf{a} \tag{5.32}$$

$$\overline{\mathbf{N}}\mathbf{v} \leq \mathbf{n} \tag{5.33}$$

$$\sum_{c \in \mathcal{C}} \gamma_c^{fr} v_c \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F. \tag{5.34}$$

Constraints (5.29) and (5.30) are the *covering* constraints associated with the pickup demands and delivery demands, respectively. Constraints (5.31)-(5.33) are the *balance*, *landing*, and *plane count* constraints described earlier. Finally, constraints (5.34) ensure that the selection of each aircraft route is integral.

5.3.2 Solution and Bounds

To put this formulation in the context of **RO**, the coefficients δ_c^{gh} arise from the *combination* of **RO** columns and *coefficient rounding* in the resulting column. Specifically, we take a linear combination of **RO** columns that corresponds to integral aircraft routes. The resulting columns are summed to give a new column with available capacity to cover some number of gateway-hub demands. The coefficients δ_c^{gh} equal 1 if, among the extreme routes that comprise c, the available capacity for commodity (g,h) exceeds the total demand for the commodity. They are set to 0 otherwise.

A composite is a combination of *extreme routes* such that the resulting *aircraft routes* are integral. In Example 23, we built a composite in which each aircraft route was represented by a single extreme route. In general, we allow each aircraft route to be specified by a combination of its extreme routes and combined with other aircraft routes to form a composite.

The number of times an aircraft route (f, r) is utilized in a composite is specified by the integer parameter, γ_c^{fr}. The composites selected by **ARM** must ensure the integrality of the *aircraft routes.* This is why we specify $\sum_{c \in C} \gamma_c^{fr} v_c$ to be integral. The more restrictive integrality requirement, $v_c \in \mathbb{Z}_+$, is not needed because integral aircraft routes can be generated with fractional composites.

We let γ_c^e denote the usage of extreme route e in composite c, where γ_c^e can be fractional. We can construct γ_c^{fr} from the extreme route usage by the relation the $\gamma_c^{fr} = \sum_{e \in E} \delta_e^{fr} \gamma_c^e$. Any solution to **ARM** is mapped back to the **RO** solution by the relation:

$$w_e = \sum_{c \in C} \gamma_c^e v_c. \tag{5.35}$$

We may also use γ_c^e to link the composite's available capacity to a particular gateway-hub demand via the relation:

$$\sum_{e \in E_P} \widehat{u}_{gh}^e \gamma_c^e \geq \delta_c^{gh} b_P^{gh}. \tag{5.36}$$

This says that if the extreme routes selected for a composite provide enough capacity, we can treat the demand as *covered* (i.e., $\delta_c^{gh} = 1$). Finally, we can map any **ARM** solution to an **ESSND-R** solution with the relation:

$$y_r^f = \sum_{c \in C} \gamma_c^{fr} v_c. \tag{5.37}$$

Lemma 26 ***ARM** and **RO** are equivalent integer programming formulations.*

Proof. (Arguments are presented in terms of *pickup* routes and apply directly to the delivery side.) Given a feasible **RO** solution, $\widehat{\mathbf{w}}$, we construct an **ARM** solution as follows. We define a single column in **ARM**, denoted by c^*, such that $\overline{\mathbf{B}}_{c^*} = \sum_{e \in E_P} \widehat{\mathbf{B}}_e \widehat{w}_e$, and where $\overline{\mathbf{A}}_{c^*}$, $\overline{\mathbf{C}}_{c^*}$ and $\overline{d}_{c^*}$ are found similarly. For the capacity-demand constraints, summing the extreme route columns yields $\overline{u}_{gh}^{c^*} = \sum_{e \in E_P} \widehat{u}_{gh}^{e} \widehat{w}_e \geq b_P^{gh}$ for the pickup capacity-demand constraints. So $\delta_{c^*}^{gh} = 1$ for all $(g,h) \in \mathcal{K}_P$. The aircraft usage within c^* is given by $\gamma_{c^*}^{fr} = \sum_{e \in E_P} \delta_e^{fr} \widehat{w}_e$, which is integral for all (f,r). We similarly define a second composite, c^{**}, for the demand side. Let, $\widehat{v}_{c^*} = 1$ and $\widehat{v}_{c^{**}} = 1$, which satisfy all constraints in **ARM** and has the same cost as the **RO** solution. Finally, $\gamma_{c^*}^{fr}\widehat{v}_{c^*} + \gamma_{c^{**}}^{fr}\widehat{v}_{c^{**}}$ is integral due to the integrality of $\gamma_{c^*}^{fr}$ and $\gamma_{c^{**}}^{fr}$, for all (f,r).

Conversely, assume we are given an **ARM** solution, $\overline{\mathbf{v}}$. We map $\overline{\mathbf{v}}$ to an **RO** solution, $\overline{\mathbf{w}}$, as in (5.35). Using the capacity relation (5.36), the capacity assigned to commodity $(g,h) \in \mathcal{K}_P$ in **RO** is:

$$\begin{aligned} \sum_{e \in E_P} \widehat{u}_{gh}^{e} \overline{w}_e &= \sum_{e \in E_P} \sum_{c \in \mathcal{C}_P} \widehat{u}_{gh}^{e} \gamma_c^e \overline{v}_c \\ &\geq \sum_{c \in \mathcal{C}_P} \overline{v}_c \delta_c^{gh} b_P^{gh} \\ &\geq b_P^{gh} \end{aligned}$$

and capacity-demand constraints (5.21) are satisfied. The delivery capacity-demand constraints (5.22) are similarly satisfied. Balance constraints (5.23) in **RO** are satisfied since $\widehat{\mathbf{B}}\overline{\mathbf{w}} = \sum_{e \in E} \sum_{c \in C} \widehat{\mathbf{B}}_e \gamma_c^e \overline{v}_c = \sum_{c \in C} \overline{\mathbf{B}}_c \overline{v}_c = \overline{\mathbf{B}}\overline{\mathbf{v}} = 0$. We similarly establish that the aircraft landing (5.24) and plane count (5.25) constraints are satisfied and that the cost of the **RO** solution is the same as the cost of the **ARM** solution. Integrality of the aircraft routes follows directly from the mapping (5.37), completing the proof. ■

If we consider a feasible solution to the LP relaxation of ARM, we can establish the following result directly from the arguments used in proving the converse of Lemma 26 by showing a feasible solution to **ARM**'s LP relaxation has a corresponding feasible solution to the **RO** LP relaxation with the same cost.

Lemma 27 *The **ARM** LP relaxation is at least as strong as that of **RO**.*

In much the same way that **RO**'s LP relaxation can be strictly greater than that of **ESSND-R**, **ARM**'s LP relaxation can be strictly greater than that of **RO**. We will explore that relationship in Section 5.4. The main result of this section follows directly from Lemmas 21 and 26 for equivalence and Lemmas 22 and 27 for strength:

Theorem 28 ***ARM*** *is equivalent (in the integer programming sense) to* ***ESSND-R*** *and its LP relaxation is at least as strong as that of* ***ESSND-R****.*

Next, we compare **ARM** to **ESSND-R** when the latter is strengthened with valid inequalities. Specifically, we explore aggregate capacity-demand constraints strengthened by Chvátal-Gomory cuts, which were introduced to the **ESSND** problem in Kim et al. [52]. As usual, we will consider the case of pickup routes, recognizing that symmetric arguments apply to the delivery side.

We refer to the set of locations (i.e., gateways and hubs) as the set N. We partition N into two non-overlapping sets, S and T, such that $S \cup T = N$. Let $[S,T]$ denote the set of routes that pass from S to T. A double-leg route might pass from T to S and from S to T and we consider it to be an element of the cut. Similarly, if the route passes from S to T and from T to S, we consider it part of the cut. We assume aircraft capacities are dependent only upon the aircraft type and we denote these capacities by u^f for all $f \in F$. Let $D_{S,T}$ be the total demand for commodities that have their origin (a gateway) in S and the destination (a hub) in T.

Any feasible solution to **ESSND-R** satisfies the following aggregate capacity-demand constraint:

$$\sum_{(f,r)\in[S,T]} u^f y_r^f \geq D_{S,T}.$$

Applying Chvátal-Gomory rounding, we obtain the inequalities:

$$\sum_{(f,r)\in[S,T]} \left\lceil \frac{u^f}{u^l} \right\rceil y_r^f \geq \left\lceil \frac{D_{S,T}}{u^l} \right\rceil \quad \text{for all } l \in F. \tag{5.38}$$

We say inequalities are *valid* if adding them to a (mixed) integer program does not affect the set of feasible integer solutions.

Lemma 29 (see Kim et al., 98) *For any $[S,T]$ cut, (5.38) are valid inequalities for the mixed integer program* ***ESSND-R****.*

Proof. For any $l \in F$, we have $\sum_{(f,r)\in[S,T]} \left\lceil \frac{u^f}{u^l} \right\rceil y_r^f \geq \sum_{(f,r)\in[S,T]} \frac{u^f}{u^l} y_r^f \geq \frac{D_{S,T}}{u^l}$. The routes that cross the cutset are integral, as are the terms $\left\lceil \frac{u^f}{u^l} \right\rceil$. Thus, the left-most term is integral and so $\sum_{(f,r)\in[S,T]} \left\lceil \frac{u^f}{u^l} \right\rceil y_r^f \geq \left\lceil \frac{D_{S,T}}{u^l} \right\rceil$. ■

In the next theorem, we establish that any solution to **ARM** (integral or fractional) has a corresponding solution that is *feasible* with respect to **ESSND-R** and the valid cutset inequalities (5.38) Moreover, these solutions have the same cost. We denote as **ESSND-Cut** the formulation **ESSND-R** with the C-G cuts (5.38) added for *all* $[S,T]$ cuts.

Theorem 30 *The **ARM** LP relaxation is at least as strong as that of **ESSND-Cut**.*

Proof. (Arguments are presented in terms of *pickup* routes and apply directly to the delivery side.) Take any composite $c \in \mathcal{C}$ that consists of aircraft routes that cross the $[S,T]$ cut. By definition, the composite cover has adequate capacity to cover some set of commodities. The parameter γ_c^{fr} specifies the *integral* number of each aircraft route included in the composite, δ_c^{gh} indicates whether the composite covers the demand for commodity (g,h), and $\mathcal{K}_{S,T}$ denotes the subset of commodities with their origin in S and destination in T. We denote the demand across the cut for the commodities covered by c as $D_{S,T}^c = \sum_{(g,h)\in\mathcal{K}_{S,T}} \delta_c^{gh} b_P^{gh}$. Because the composite ensures a feasible flow for these commodities, the aggregate capacity-demand constraint is satisfied on a portion of the cutset:

$$\sum_{(f,r)\in[S,T]} u^f \gamma_c^{fr} \geq D_{S,T}^c.$$

Because aircraft usage in each composite is integral (i.e., $\gamma_c^{fr} \in \mathbb{Z}_+$), γ_c^{fr} must also satisfy the C-G cuts because the cuts are *valid* (Lemma 29):

$$\sum_{(f,r)\in[S,T]} \left\lceil \frac{u^f}{u^l} \right\rceil \gamma_c^{fr} \geq \left\lceil \frac{D_{S,T}^c}{u^l} \right\rceil \quad \text{for all } l \in F. \tag{5.39}$$

Let our choice of composites be specified by $\widehat{\mathbf{v}}$. Then for all $l \in F$:

$$\sum_{(f,r)\in[S,T]} y_r^f \left\lceil \frac{u^f}{u^l} \right\rceil = \sum_{(f,r)\in[S,T]} \left(\sum_{c\in\mathcal{C}} \gamma_c^{fr} \widehat{v}_c \right) \left\lceil \frac{u^f}{u^l} \right\rceil$$

$$
\begin{aligned}
&= \sum_{c\in C}\left(\sum_{(f,r)\in[S,T]}\left\lceil\frac{u^f}{u^l}\right\rceil \gamma_c^{fr}\right)\widehat{v}_c \\
&\geq \sum_{c\in C}\left(\left\lceil\frac{D_{S,T}^c}{u^l}\right\rceil\right)\widehat{v}_c \\
&\geq \left\lceil\frac{\sum\limits_{c\in C}\widehat{v}_c D_{S,T}^c}{u^l}\right\rceil,
\end{aligned}
$$

where the first inequality follows from (5.39). Next, the quantity being rounded can be simplified as follows:

$$
\begin{aligned}
\frac{\sum\limits_{c\in C}\widehat{v}_c D_{S,T}^c}{u^l} &= \frac{\sum\limits_{c\in C}\sum\limits_{(g,h)\in\mathcal{K}_{S,T}}\widehat{v}_c\delta_c^{gh}b^{gh}}{u^l} \\
&\geq \frac{\sum\limits_{(g,h)\in\mathcal{K}_{S,T}} b^{gh}}{u^l} \\
&= \frac{D_{S,T}}{u^l}
\end{aligned}
$$

where the inequality results from the fact that $\sum\limits_{c\in C}\widehat{v}_c\delta_c^{gh} \geq 1$ for all $(g,h)\in\mathcal{K}$. This establishes the desired result. ■

5.4 Examples of Strict Improvement in the Bounds

In the previous section, we established that **ESSND-R**, **RO**, and **ARM** are equivalent formulations with respect to integral aircraft route solutions. We further established that **ARM** is at least as strong as **RO**, which is at least as strong as **ESSND-R**. In this section we explore the case when one formulation is strictly stronger than another, that is, we can demonstrate that the bound provided by the LP relaxation of one formulation is better than the LP relaxation of the other. We do this through two examples. First, using a simple two-node, two-aircraft network, we investigate the properties of solutions for **ESSND-R**, **RO**, and **ARM**. Second, using a more complex (yet simple) single-hub network, we examine the general effect of formulations on the bounds provided by their LP relaxations.

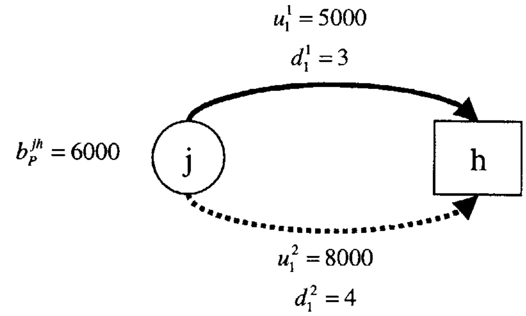

Figure 5-6: Simple two-node network demonstrating formulation strength

5.4.1 Two-Node Example

Consider the network shown in Figure 5-6. We have a single route from gateway j to hub h and the route can be flown by two aircraft types. The first has a capacity of 5000 packages and a cost of 3. The second has a capacity of 8000 packages and a cost of 4. Our objective is to move all 6000 packages at minimum cost from the gateway to the hub.

The **ESSND-R** formulation (excluding the landing, plane count, and aircraft balance constraints) is

$$\begin{aligned} \min \quad & 3y_1^1 + 4y_1^2 \\ \text{subject to:} \quad & x_1^{jh} - 5000y_1^1 - 8000y_1^2 \leq 0 \\ & x_1^{jh} = 6000 \\ & y_1^1, y_1^2 \in \mathbb{Z}_+. \end{aligned}$$

In this simple case with only a single commodity and single-leg routes, we could simplify this formulation but will keep it in the **ESSND** form for the purpose of exposition. The optimal solution to the LP relaxation can be found by flowing all packages on the aircraft with the lowest cost per unit of capacity. The cost per unit capacity is $\frac{3}{5000}$ for the first aircraft route and $\frac{1}{2000}$ for the second. Thus, the optimal solution is $x_1^{jh} = 6000$, $y_1^1 = 0$, and $y_1^2 = 0.75$, with a total cost of 3. Note that the optimal integer solution is to fly the type 2 aircraft, or $y_1^2 = 1$, with a cost of 4.

Next, we construct the **RO** formulation. Because we have two *single-leg* routes, the **RO** formulation contains only two extreme routes, one for each of the original aircraft routes. Calculating the available capacities of our two extreme routes, we have $\widehat{u}^1_{jh} = \min\{b^{jh}_P, u^1_1\} = 5000$ and $\widehat{u}^2_{jh} = \min\{b^{jh}_P, u^2_1\} = 6000$. **RO** is given by

$$\begin{aligned} \min\ & 3w_1 + 4w_2 \\ \text{subject to:}\quad & 5000w_1 + 6000w_2 \geq 6000 \\ & w_1, w_2 \in \mathbb{Z}_+. \end{aligned}$$

We apply the same reasoning as before to obtain the optimal solution to **RO**'s LP relaxation. The cost per unit of *available* capacity is $\frac{3}{5000}$ for the first extreme route and $\frac{2}{3000}$ for the second so we choose to fly the first aircraft in sufficient quantity to cover the entire demand. Thus, the optimal solution for the LP relaxation is $w_1 = 1.2$ with a cost of 3.6.

The essence of the **RO** formulation improvement is as follows. A fractional aircraft solution in **ESSND-R** avoided being charged the cost of flying empty aircraft. By using extreme routes with available capacities that are lower than the capacities of the original routes, **RO** reclaims part of the empty portion of aircraft routes. That is, **RO** places total aircraft route cost on the available capacity rather than on the actual capacity.

In our example, the **ESSND-R** fractional solution of $\frac{3}{4}$ type 2 aircraft (and no type 1 aircraft) **cannot** be represented in **RO**. The maximum available capacity for the type 2 aircraft is $\widehat{u}^2_{jh} = 6000$. By selecting $\frac{3}{4}$ of the type 2 aircraft in **RO**, we would only be able to flow 4500 packages, which is infeasible. The optimal integer solution to **RO** is $w_2 = 1$ with a total cost of 4.

We reformulate **RO** by scaling the capacity-demand constraints:

$$\begin{aligned} \min\ & 3w_1 + 4w_2 \\ \text{subject to:}\quad & \frac{5}{6}w_1 + 1w_2 \geq 1 \\ & w_1, w_2 \in \mathbb{Z}_+. \end{aligned}$$

The second extreme route is itself a composite because it covers the entire demand. We call this composite 1. We build a second composite variable by doubling the first column. We'll call this composite 2. The mapping of each composite to aircraft routes (γ_c^{fr}) is given by $\gamma_1^{1,1} = 0$ and $\gamma_1^{1,2} = 1$ for composite 1 and $\gamma_2^{1,1} = 2$ and $\gamma_2^{1,2} = 0$ for composite 2. **ARM** is then:

$$\min\ 4v_1 + 6v_2$$

$$\text{subject to:}\quad v_1 + v_2 \geq 1$$

$$v_1\gamma_1^{1,1} + v_2\gamma_2^{1,1} \in \mathbb{Z}_+$$

$$v_1\gamma_1^{1,2} + v_2\gamma_2^{1,2} \in \mathbb{Z}_+.$$

The first integrality requirement reduces to $2v_2 \in \mathbb{Z}_+$ while the second integrality requirement reduces to $v_1 \in \mathbb{Z}_+$.

The optimal solution to the LP relaxation of **ARM** is $v_1 = 1$ with a cost of 4.0. The bound provided by **ARM** is tighter than the bound provided by **RO**. In fact, the bound could not be any tighter as the optimal solution to this LP relaxation is integral and, hence, it is the optimal solution to the integer program.

Recall that the fractional solution to **RO** used $\frac{6}{5}$ of type 1 aircraft and *no* type 2 aircraft. There is no corresponding feasible solution in **ARM**. Any **ARM** solution that uses *no* type 2 aircraft is forced to use at least **two** type 1 aircraft. This accounts for **ARM** being stronger than **RO** in this example.

In going from **RO** to **ARM**, we observe a phenomenon similar to what we observed going from **ESSND-R** to **RO**. With the demand from g to h larger than the capacity of the first fleet type, flying only the first fleet type on this route requires using more than one aircraft. But the second aircraft we fly will be partially utilized as the model would decide not to fly excess capacity due to the extra cost. By rounding the capacity of the two type 1 aircraft to have a capacity equal to the (g, h) demand, we force the solution to use (and pay for) excess capacity in the solution.

To summarize this phenomenon, the presence of excess capacity in aircraft routes increases the opportunity for fractionality in the solution to the **ESSND-R** LP relaxation. By defining a model that uses extreme routes, we are able to absorb part of the excess capacity through coeffi-

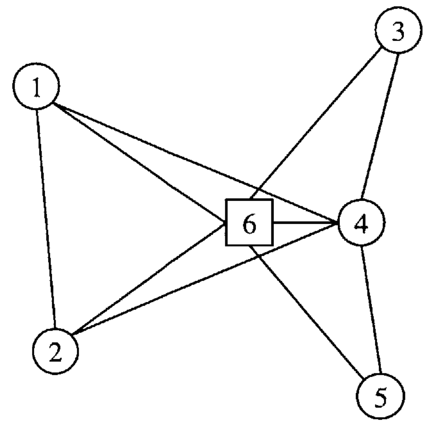

Figure 5-7: Single-hub network for demonstrating the strength of **ESSND**, **RO**, and **ARM**

cient reduction thereby removing some of this fractionality. In **RO**, however, this absorption is accomplished only on individual aircraft routes whose capacity exceeds one of its gateway-hub demands. When individual aircraft routes do not cover their gateway-hub demands, we create composite variables to cover these demands. The process of combining routes may, however, result in excess capacity, which we can absorb by reducing the capacity-demand coefficients and reduce the fractionality even further.

5.4.2 Single Hub Example

To provide a more intuitive sense of the formulations and a more complete picture of their computational behavior, we present a system containing only a single hub. Without package flow costs, there is no benefit in switching packages between planes at an intermediate gateway location when all packages are bound for the same hub. Thus, we do not need to consider ramp transfer composite variables in this example, allowing us to formulate and solve the three models presented in this chapter.

Consider the network depicted in Figure 5-7. On this network we have timing restrictions for pickup and delivery at gateways and timing restrictions for the hub sort. We use three fleet types, ranging in capacity from 8000 packages to 10000 packages. One fleet type is restricted to flying only single-leg routes.

Given the time windows and the speed and range of each aircraft, we identify 11 pickup

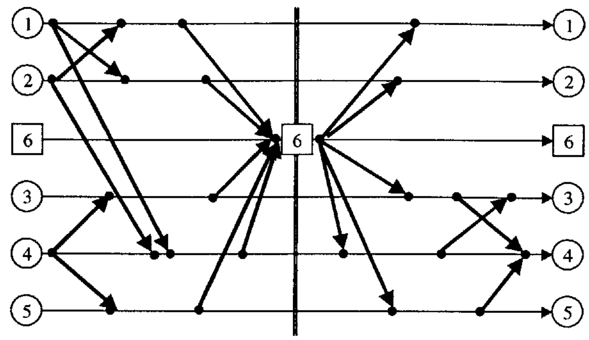

Figure 5-8: Time-space network showing feasible routes for single-hub example

routes and 8 delivery routes (shown in the time-space network in Figure 5-8). On the pickup side, three routes can be flown by all three aircraft types, six routes can be flown by two aircraft types, and two routes can be flown by only one aircraft type (this yields a total of 19 aircraft routes for pickup). On the delivery side, three routes can be flown by all three aircraft types, two routes can be flown by two aircraft types, and three routes can only be flown by a single aircraft type (this yields a total of 16 aircraft routes for delivery).

Table 5.1 shows the size of each of the three formulations: **ESSND**, **RO**, and **ARM**. The reduction in the number of variables from **ESSND** to **RO** stems from the removal of the package flow variables. In this example, many of the gateway-hub demands are small enough so that any route incident to those gateways is guaranteed to have sufficient capacity to carry the demand. In the **RO** formulation, such routes are not required to be split into *extreme routes* because, for double-leg pickup routes from i to j to h, $b_P^{ih} + b_P^{ih} \leq u_r^f$ and for single-leg pickup routes from i to h, $b_P^{ih} \leq u_r^f$.

One gateway has delivery demand that exceeds the capacity of all aircraft types. All delivery routes incident to this gateway are single-leg routes and, once again, the **RO** formulation does not require that these single-leg aircraft routes be represented with two extreme routes. Hence, the number of decision variables in **RO** contains one decision variable for each aircraft route. The number of constraints in **RO** has been reduced from 53 to 34, which results from the removal of *forcing* constraints and replacing the demand constraints in **ESSND** with the capacity-demand constraints in **RO**.

Going from **RO** to **ARM**, the number of constraints is the same, as the capacity-demand

	ESSND	RO	ARM
Rows	53	34	34
Cols	67	39	42
Nonzeros	274	231	255

Table 5.1: Size of formulations for single-hub problem

	ESSND	RO	ARM
LP Relaxation Solution	10663.037	23154.683	28474.014
IP Solution	28474.014	28474.014	28474.014
Nodes in B-B tree	781	111	1
Gap ($\frac{z_{IP}^* - z_{LP}^*}{z_{IP}^*}$)	0.6255	0.1868	0.0000

Table 5.2: Solution summary for **ESSND**, **RO**, and **ARM** applied to single hub example

constraints in **RO** are transformed, through column combination and rounding-down, to the covering constraints in **ARM**. All other constraints remain the same. The number of columns increases by three, as **ARM** now includes composite variables versus extreme route variables.

This example has **ARM** smaller than **ESSND**, at least as measured by row and column counts. In general, this is not the case. The number of columns in **ARM** can explode, as we create combinations of aircraft routes that will cover commodity demands.

The solutions for the three models were generated using XPRESS-MP v.10 on a 300 Mhz Pentium PC. The results are summarized in Table 5.2. These are consistent with what we have established with respect to the strength of the three models. As we move from **ESSND** to **ARM**, we see the LP relaxation gives a better approximation of the optimal integer solution. In fact, for this example, the LP relaxation is integral and, therefore, the optimal solution the integer program.

5.5 Optimality of ARM with Restricted Composite Set

In this section, we look at how the composites we include in **ARM** are affected by the "non-ramp transfer" assumption along with a particular operational assumption we have not yet considered. Under these assumptions, the set of composites we build in the restricted **ARM**, $\mathcal{C}^r$ (see Chapter 3), yields the *optimal* solution to **ESSND**. Finally, we address the issue of ramp transfer and how our UPS implementation of **ARM** creates ramp transfer composite variables.

In the transformation from **RO** to **ARM**, we generated composite variables from combinations of extreme routes. That is, a single aircraft route (f, r) used in a composite, c, is built from a convex combination of the (f, r) extreme routes. The total usage of (f, r) in the composite (say it is used two or more times), which we denote by γ_c^{fr}, arises from the more general combination:

$$\gamma_c^{fr} = \sum_{e \in E} \delta_e^{fr} w_e \in \mathbb{Z}_+$$

for some set of weights, w_e, $e \in E$, that are not required to be integral. The shear size of the composite set might present some difficulty in constructing the composites. The following operational assumption eases this difficulty:

> **Operational Assumption:** *Double-leg routes shall only be used if they cover at least one of their gateway-hub demands.*

This assumption has a strong effect on the set of composites that we construct for **ARM**. We explore its effect on the extreme routes in the **RO** formulation followed by examining its impact on **ARM**.

Lemma 31 *Under the* ***operational assumption****, a double-leg aircraft route may only be represented by its extreme routes, not convex combinations of its extreme routes. Furthermore, these extreme routes must obey the operational assumption.*

Proof. If a double-leg route has only one extreme route, its capacity exceeds the total demand of its gateway-hub pairs, which satisfies the operational assumption. So we consider only the case when a double-leg aircraft route has two extreme routes. Any (non-trivial) convex combination of the two extreme routes provides available capacity to the gateway-hub pairs that is strictly less than the gateway-hub demand. Thus, we can only represent an aircraft route by its extreme routes. If an extreme route does not satisfy the operational assumption, then representing the aircraft route with this extreme route gives an aircraft route that doesn't satisfy the operational assumption. ■

The implication of this Lemma to **RO** is the following: we can replace the requirement for integral aircraft routes (which are constructed from extreme routes) with the requirement for integral extreme routes. That is, replace $\sum_{e \in E} \delta_e^{fr} w_e \in \mathbb{Z}_+$, for all (f, r), with the requirement $w_e \in \mathbb{Z}_+$, for all $e \in E^*$. E^* denotes the subset of extreme routes that satisfy the operational assumption.

The implication to **ARM** is that composites cannot be built from anything but the individual extreme routes in E^*. Otherwise the operational assumption is violated. Combined with the earlier proofs of equivalence of the (mixed) integer programming formulations **ESSND-R** and **ARM**, this gives the following result:

Theorem 32 *Using composites constructed using integral extreme routes from the set E^*, **ARM** yields the optimal integer solution to **ESSND** under the operational assumption.*

In Chapter 3, we described procedures for constructing non-ramp transfer composites in exactly this manner. When we remove the artificial "no ramp transfer" assumption from the beginning of this chapter, we must consider composites that represent ramp transfers. Due to strict customer service and timing requirements and because of the pre-assignment of gateways to a limited number of hub territories, there is a natural limit on the number of ramp transfer composites that we need to consider. We simply generate *all* of them explicitly (as specified in Chapter 3) and add them to the model. The total number of composites (both ramp transfer and non-ramp transfer) is "small enough" to include all composites in **ARM** and solve. The tight bounds provided by **ARM**'s LP relaxation help us quickly find very good integer solutions.

If we remove the operational assumption and consider more complicated interactions between extreme routes, we expect the number of composites in C^r to increase. In this case, we can store composites in an off-line oracle and work with a restricted master problem in the context of branch-and-price (see Barnhart et al. [16]). The explicit pricing mechanism used by the oracle would take advantage of the fact that all composites are built from the same set of aircraft routes. Each aircraft route has dual information that is independent of the composite(s) in which it is contained. Then the reduced cost of each composite can be found from its component aircraft routes' dual information combined with composite-specific dual information.

5.6 Summary

We have described the relationship between the Aircraft Routing Model (**ARM**) and the original Express Shipment Service Network Design (**ESSND**) formulation. We have established **ARM** to be at least as strong as **ESSND** through an intermediate model (**RO**). The **RO** model provides an intuitive means for understanding the composite variable formulation and for establishing the equivalence and relative strength of the models. Through the equivalence of **ARM** and **ESSND**, we have shown that under certain reasonable assumptions, the version of **ARM** with the restricted set of composites (as described in Chapter 3 and used in the UPS case study in Chapter 4) yields the optimal integer solution to **ESSND**.

Chapter 6

General Interpretations of Composite Variable Formulations

In Chapter 2, we presented the general form of the Express Shipment Service Network Design (**ESSND**) problem and two equivalent formulations based on a single decomposition strategy. In the third chapter, we presented the idea of composite variable formulations as a way to remove the package flow variables and constraints by capturing them implicitly in the decision variables of the new formulation. The computational results presented in Chapter 4 were based on an implementation of the Aircraft Routing Model (**ARM**) using a restricted set of composite variables. Chapter 5 established the link between **ESSND** and **ARM** and the relative strength of the two formulations.

In this chapter we continue the discussion of this link but do so by looking at aspects that will contribute to the generalization of composite variable formulations. We interpret the exact version of **ARM** as a Dantzig-Wolfe decomposition of **ESSND**. In doing so, we establish a theoretical framework for improving our heuristic **ARM** solution and for establishing lower bounds on the optimal **ESSND** solution. In addition, we present an interpretation of **ARM** that relates the composite variables to cutting planes in the dual problem. Finally, we explore the use of composite variable formulations in the context of a general network design problem known as the Pure Fixed Charge Transportation Problem.

6.1 Dantzig-Wolfe Interpretation of ARM

We begin by re-stating **ESSND** and the Dantzig-Wolfe decomposition that led to the origin (i.e., supercommodity) and origin-destination formulations presented in Chapter 2. We then present an alternative decomposition in which the master problem contains only design (i.e., aircraft route) variables. The resulting master problem is simply **ARM** and the subproblem generates integer extreme point solutions that are related to our composite variables. From this decomposition we are able to derive lower bounds that give us an alternative to the weak lower bounds produced by the **ESSND** LP relaxation.

6.1.1 ESSND Problem Formulation

We begin by re-stating notation introduced in earlier chapters:

Sets

F	Set of fleet types
H	Set of hubs
G	Set of gateways
R^f	Set of routes flown by fleet type $f \in F$
A	Set of arcs in the derived time-space network
N	Set of nodes in derived time-space network
$\mathcal{K}$	Set of commodities to flow through the network

Decision Variables and Costs

$\mathbf{y}$	Vector of aircraft route variables $y_r^f \in \mathbb{Z}_+$, $r \in R^f$, $f \in F$
$\mathbf{x}$	Vector of package flow variables $x_{ij}^k \geq 0$, $(i,j) \in A$, $k \in \mathcal{K}$
$\mathbf{d}$	Vector of aircraft route costs d_r^f, $r \in R^f$, $f \in F$

Constraint Matrices

$\mathbf{U}$	Mapping of aircraft route capacities u_r^f to arcs in A
$\mathcal{N}^k$	Node-arc incidence matrix for commodity $k \in \mathcal{K}$
$\mathbf{B}$	Balance constraint matrix

A Landing (Arrival) constraint matrix

N Plane count (Number) constraint matrix

Right-Hand-Side Data

$\mathbf{b}^k$ Demand vector for commodity k with components b_i^k, $i \in N$

a Vector of hub arrival capacities with components a_h, $h \in H$

n Vector of fleet sizes with components n_f, $f \in F$

Under the assumption that package flow costs are zero and the assumption that O-D commodities have been pre-assigned to hubs, the general form of the problem (**ESSND**) is given by:

$$\min \mathbf{d}'\mathbf{y} \tag{6.1}$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} \mathbf{x}^k - U\mathbf{y} \leq \mathbf{0} \tag{6.2}$$

$$\mathcal{N}^k \mathbf{x}^k = \mathbf{b}^k \quad k \in \mathcal{K} \tag{6.3}$$

$$\mathbf{B}\mathbf{y} = \mathbf{0} \tag{6.4}$$

$$\mathbf{A}\mathbf{y} \leq \mathbf{a} \tag{6.5}$$

$$\mathbf{N}\mathbf{y} \leq \mathbf{n} \tag{6.6}$$

$$y_r^f \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F \tag{6.7}$$

$$x_{ij}^k \geq 0 \quad k \in \mathcal{K},\ (i,j) \in A. \tag{6.8}$$

Forcing constraints (6.2) ensure the flow assigned to each arc is no more than the capacity assigned to the arc. Constraints (6.3) ensure *conservation of flow* for each commodity. Constraint (6.4) ensure *aircraft balance*, constraints (6.5) enforce *landing capacities* at the hubs, and constraints (6.6) ensure the solution satisfies *plane counts* for each fleet type.

6.1.2 Decomposition of Package Flow Variables

A decomposition strategy introduced in Kim et al. [52] is geared at reducing the millions of conservation of flow constraints (6.3) in **ESSND**. Jones et al. [48] have previously applied similar ideas to multicommodity network flow problems.

Let $\mathbf{x}_e^k$, $e \in \mathcal{E}^k$, be the extreme points of the network flow polyhedra defined by (6.3) and (6.8) for commodity k (we do not consider extreme rays, i.e., negative cost cycles, because the underlying network has nonnegative arc costs). Each $\mathbf{x}_e^k$ is a vector of length $|A|$ with each component corresponding to a flow on an arc $(i,j) \in A$. We define λ_e^k to be the weight associated with extreme point $\mathbf{x}_e^k$, $e \in \mathcal{E}^k$. The master problem ($\mathbf{ESSND}_{MCF}$), stated in terms of decision variables λ_e^k, is:

$$\min \mathbf{d}'\mathbf{y} \tag{6.9}$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} \sum_{e \in \mathcal{E}^k} \mathbf{x}_e^k \lambda_e^k - U\mathbf{y} \leq \mathbf{0} \tag{6.10}$$

$$\mathbf{B}\mathbf{y} = 0 \tag{6.11}$$

$$\mathbf{A}\mathbf{y} \leq \mathbf{a} \tag{6.12}$$

$$\mathbf{N}\mathbf{y} \leq \mathbf{n} \tag{6.13}$$

$$\sum_{e \in \mathcal{E}^k} \lambda_e^k = 1 \quad k \in \mathcal{K} \tag{6.14}$$

$$y_r^f \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F \tag{6.15}$$

$$\lambda_e^k \geq 0 \quad e \in \mathcal{E}^k,\ k \in \mathcal{K}. \tag{6.16}$$

At each iteration of the Dantzig-Wolfe decomposition algorithm, the master problem includes a subset of the total number of columns. The reduced model is referred to as the *restricted master problem* (RMP). An optimal solution to the RMP has dual information that is used to determine if any columns not in the RMP have negative reduced cost. We define the dual variables as follows:

- $\boldsymbol{\pi}^u$ Forcing constraint duals with components $\pi_{ij}^u \leq 0$, $(i,j) \in A$
- $\boldsymbol{\pi}^b$ Balance constraint duals with components π_{if}^b, $f \in F$, $i \in G \cup H$
- $\boldsymbol{\pi}^a$ Landing constraint duals with components $\pi_h^a \leq 0$, $h \in H$
- $\boldsymbol{\pi}^n$ Plane count constraint duals with components $\pi_f^n \leq 0$, $f \in F$
- $\boldsymbol{\sigma}$ Convexity constraint duals with components σ_k, $k \in \mathcal{K}$

We use the subproblem to find extreme points of the polyhedra defined by (6.3) and (6.8). Hence, this decomposition uses only the duals associated with package flow constraints, which include the forcing constraint duals, $\boldsymbol{\pi}^u$, and the convexity constraint duals, $\boldsymbol{\sigma}$. The reduced cost for the column corresponding to flowing commodity k using extreme point e is given by $\overline{c}_e^k = \left(\mathbf{c}^k - \boldsymbol{\pi}^u\right)' \mathbf{x}_e^k - \sigma_k$.

We generate extreme points by assigning an arc cost of $c_{ij}^k - \pi_{ij}^u$ to each flight arc in the network. Under our assumptions that $c_{ij}^k = 0$ for all $(i,j) \in A$ and $k \in K$, we solve the following network flow subproblem ($\mathbf{SUB}_{MCF(k)}$):

$$\min \; -(\boldsymbol{\pi}^u)' \mathbf{x}^k \tag{6.17}$$

$$\begin{aligned} \text{subject to} \quad \mathcal{N}^k \mathbf{x}^k &= \mathbf{b}^k & (6.18)\\ x_{ij}^k &\geq 0 \;\; \forall (i,j) \in A. & (6.19) \end{aligned}$$

If the optimal solution to $\mathbf{SUB}_{MCF(k)}$ has cost less than σ_k, the extreme point has negative reduced cost (i.e., $-(\boldsymbol{\pi}^u)'x^k - \sigma_k < 0$) and is added to the restricted master problem.

As shown in Chapter 2 and in Jones, et al. [48], the structure of the extreme points $\mathbf{x}_e^k$ depends upon the definition of the commodities. When working with origin-destination commodities, the demand vector, $\mathbf{b}^k$, contains two nonzero elements: one corresponding to the demand's source at location $O(k)$ and one corresponding to the demand's sink at location $D(k)$. Consequently, the extreme point $\mathbf{x}_e^k$ represents a flow of b^k units of commodity k along the shortest path (with respect to $-\boldsymbol{\pi}^u$) from origin to destination.

Alternatively, we may define *supercommodities*. Each supercommodity is a grouping of all O-D commodities that share a common origin. With K^s denoting the O-D commodities in supercommodity s, the number of nonzero elements in the subproblem demand vector is $|K^s|+1$: one for the supercommodity origin and one for each of its O-D commodity destinations. Thus, the subproblem generates an extreme point that represents a *group* of shortest path flows from the common origin to each of the destinations.

The manner in which we solve the supercommodity subproblems depends upon the arc costs. In general, this subproblem is solved by finding the shortest path for each O-D commodity contained in the supercommodity using commodity-specific arc costs. Because each O-D commodity is contained in a single supercommodity, the number of shortest path problems that need to be solved at each iteration of the decomposition algorithm is the same as when we work directly with O-D commodities.

In the special case when each arc cost within a supercommodity is *independent* of O-D commodity, the extreme point is a shortest path *tree*. The worst-case complexity of finding the shortest path tree is the same as finding a single shortest path. In this case, the supercommodity formulation requires less work (in the worst-case) to generate columns at each iteration of the decomposition algorithm than the O-D formulation.

6.1.3 Decomposition with Design-Only Master Problem

By decomposing **ESSND** with respect to package flow variables, the master problem retains the forcing constraints. As we have seen, these constraints induce fractionality in the LP relaxation. This problem is amplified by the aircraft balance constraints, which propagate local fractionality throughout the network. To overcome this, we decompose **ESSND** to uncouple the forcing constraints from the aircraft balance constraints. That is, we dualize both the forcing constraints (6.2) and the conservation of flow constraints (6.3). The resulting master problem will contain only design variables, with all package flow variables and constraints isolated in the subproblem. This decomposition yields a subproblem that is a Network Loading Problem (**NLP**).

Let $(\mathbf{x}_c, \mathbf{y}_c)$, $c \in \mathcal{C}^*$, be the set of extreme points of the polyhedron defined by (6.2), (6.3), (6.7), and (6.8). Each extreme point is a vector containing $|A|$ components in $\mathbf{x}_c$ and $\sum_{f \in F} |R^f|$ components in $\mathbf{y}_c$. We define v_c as the weight associated with extreme point $(\mathbf{x}_c, \mathbf{y}_c)$, $c \in \mathcal{C}^*$. The master problem ($\mathbf{ESSND}_{NLP}$), stated in terms of v_c, is given by:

$$z^*_{NLP} = \min \sum_{c \in \mathcal{C}^*} [\mathbf{d}'\mathbf{y}_c]\, v_c \tag{6.20}$$

$$\text{subject to} \quad \sum_{c \in \mathcal{C}^*} [\mathbf{B}\mathbf{y}_c]\, v_c = 0 \tag{6.21}$$

$$\sum_{c \in C^*} [\mathbf{A}\mathbf{y}_c] v_c \leq \mathbf{a} \tag{6.22}$$

$$\sum_{c \in C^*} [\mathbf{N}\mathbf{y}_c] v_c \leq \mathbf{n} \tag{6.23}$$

$$\sum_{c \in C^*} v_c = 1 \tag{6.24}$$

$$\sum_{c \in C^*} y_c^{fr} v_c \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F. \tag{6.25}$$

The constraints include *aircraft balance* (6.21), *landing capacity* (6.22), *plane count* (6.23), and *convexity* (6.24). *Integrality* is required on the aircraft routes (6.25), not the decision variables, v_c. The dual variables corresponding to constraints (6.21)-(6.23) are $\boldsymbol{\pi}^b$, $\boldsymbol{\pi}^a \leq \mathbf{0}$, and $\boldsymbol{\pi}^n \leq \mathbf{0}$, as defined earlier. The single dual variable for the convexity constraint (6.24), which we denote by σ, is unrestricted in sign.

The subproblem is a minimization over $\mathbf{x}$ and $\mathbf{y}$. The master problem, however, contains *no* constraints with $\mathbf{x}$ and, therefore, no dual information pertaining to package flows. This, along with the standing assumption that package flow costs are zero, yields a subproblem whose objective function is in terms of $\mathbf{y}$. In other words, the subproblem selects the minimum (reduced) cost set of aircraft routes for which a *feasible* package flow exists. Using the same notation for dual variables, this subproblem ($\mathbf{SUB}_{NLP}$) is:

$$z_{SUB}^* = \min \left[\mathbf{d}' - \left(\boldsymbol{\pi}^b\right)' \mathbf{B} - (\boldsymbol{\pi}^a)' \mathbf{A} - (\boldsymbol{\pi}^n)' \mathbf{N} \right] \mathbf{y} \tag{6.26}$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} \mathbf{x}^k - \mathbf{U}\mathbf{y} \leq 0 \tag{6.27}$$

$$\mathcal{N}^k \mathbf{x}^k = \mathbf{b}^k \quad k \in \mathcal{K} \tag{6.28}$$

$$x_{ij}^k \geq 0 \tag{6.29}$$

$$y_r^f \in \mathbb{Z}_+. \tag{6.30}$$

Any solution to this subproblem is a *complete* solution in the sense that a feasible flow exists for *all* commodities in the network (but not feasible for the aircraft routes as the subproblem does not contain aircraft balance, landing, and plane count constraints). The extreme points used in the master problem are complete solutions. The master problem selects the combination of complete solutions that results in the minimum cost set of integral aircraft routes.

Our goal is to have a model that does **not** require complete solutions. To do so, we need to consider changes to both the master problem and the subproblem. In the master problem we change the single convexity constraint to one convexity constraint for each commodity:

$$\sum_{c \in \mathcal{C}^*} \delta_c^k v_c = 1 \quad \forall k \in \mathcal{K}. \tag{6.31}$$

This new master problem is identical to the set partitioning form of **ARM** defined in Chapter 3. We have now transformed **ESSND** to **ARM** without the intermediate routes-only model, **RO**.

This change to the master problem has no effect other than having a convexity constraint for each commodity $k \in \mathcal{K}$. The indicator δ_c^k equals 1 if extreme point $(\mathbf{x}_c, \mathbf{y}_c)$ ensures a feasible flow (i.e., *cover*) for commodity k. Any extreme point generated by the network loading subproblem covers *all* commodity flows. That is, $\delta_c^k = 1$ for all $k \in \mathcal{K}$ and (6.31) is simply $|\mathcal{K}|$ copies of the original convexity constraint (6.14).

6.1.4 Separability of the Network Loading Subproblem

The real benefit of the $\mathbf{ESSND}_{NLP}$ decomposition lies in the (possible) separability of the subproblem. In the case when separability does not exist, the subproblem is a network loading problem and is NP-Hard (see Magnanti et al. [58]). However, our computational experience (see Chapter 4) demonstrates that without the aircraft balance constraints, finding the (optimal) integer solution is easier. Nonetheless, the subproblem is still a network loading problem. In this section, we look at three ways to either exploit or create separability in the subproblem: separating a complete *solution* (extreme point) of the network loading subproblem, separating the *subproblem* a priori, and exploiting problem structure to create localized solutions.

Separating (Disaggregating) Extreme Point Solutions

A *complete solution* to $\mathbf{SUB}_{NLP}$ covers all commodities. At a minimum, we can determine whether this solution is separable by commodity flows. That is, we try to divide the routes in this extreme point solution such that each subset of aircraft routes provides sufficient capacity for a subset of the commodities (i.e., each such subset of routes is a *composite* that *covers* the subset of commodities). Instead of placing a single column (corresponding to the extreme point solution) in the restricted master, we place one column for each subset of aircraft routes. The coverage of each new column is indicated in the master problem by setting $\delta_c^k = 1$ appropriately. While we are still required to solve the complete network loading subproblem at each iteration of the decomposition algorithm, these *subnetwork* columns give the master problem more flexibility in constructing solutions. The benefit of this type of disaggregation is demonstrated in Jones et al. [48] for multicommodity flow problems.

Separating the Network Loading Subproblem

An even stronger approach is to exploit the separability of the network loading subproblem *a priori*. If portions of the network do not interact with each other, in terms of both routes and commodities, then the network loading subproblem will have a block angular structure. That is, we can separate both the forcing constraints (6.27) and the conservation of flow constraints (6.28) into smaller network loading problems. Solving a set of these smaller problems is easier than solving the full network loading problem. The solution to each of these smaller subproblems can be represented in the master problem as a single column or we can try to disaggregate the extreme point solutions as described above. The primary difficulty, however, is identifying if (and how) the network loading subproblem separates.

Creating Localized Extreme Points

While it might not be possible to separate the network loading subproblem into subnetworks, problem-specific characteristics might provide limits on the complexity of the interactions between commodities and between aircraft routes. In the case of **ARM**, the *operational assumption* introduced in Chapter 5 allows us to build composites of limited complexity. While a given aircraft route might interact with many other routes, it will only interact with a very limited number of routes at the same time. This limits the total number of composite variables that we need to consider in the master problem (i.e., **ARM**). So this represents another form of disaggregation in which we do not need to solve the entire network loading subproblem at each iteration of the decomposition algorithm.

6.1.5 Lower Bounds

As noted in Chapter 2, the lower bounds provided by the **ESSND** LP relaxation are weak, making the search for a good integer solution and proof of optimality difficult. A consequence of the Dantzig-Wolfe framework is the possibility of deriving an alternative lower bound on **ESSND**. Our approach is to solve the **ARM** LP relaxation and use its duals in the objective function of the subproblem, $\mathbf{SUB}_{NLP}$. The subproblem objective value and the **ARM** objective value are combined to yield a lower bound for the optimal (integer) solution to **ESSND**. This bounding method capitalizes upon the strength of **ARM**'s LP relaxation and the meaningful dual information that results from solving **ARM**. These bounds are not necessarily better than those provided by the **ESSND** LP relaxation; rather, they are simply an alternative.

Bounds from the ARM LP Relaxation

We denote the implementation of **ARM** with a restricted set of composites simply as **ARM**, its LP relaxation as $\mathbf{ARM}_{LP}$, and the dual of the relaxation as $\mathbf{ARM}_{D}$. The commodity set $\mathcal{K}$ is defined by gateway-hub demands and the set $\mathcal{C}^r$ is the set of composites over which **ARM** is optimized (which might not be the full set of composite variables). $\mathbf{ARM}_{LP}$ is given by:

$$
\begin{aligned}
z^*_{ARM_{LP}} = \min \; & \sum_{c \in \mathcal{C}^r} \left[\mathbf{d}'\mathbf{y}_c\right] v_c \\
\text{subject to} \quad & \sum_{c \in \mathcal{C}^r} \left[\mathbf{B}\mathbf{y}_c\right] v_c = \mathbf{0} \\
& \sum_{c \in \mathcal{C}^r} \left[\mathbf{A}\mathbf{y}_c\right] v_c \leq \mathbf{a} \\
& \sum_{c \in \mathcal{C}^r} \left[\mathbf{N}\mathbf{y}_c\right] v_c \leq \mathbf{n} \\
& \sum_{c \in \mathcal{C}^r} \delta_c^k v_c = 1 \quad k \in \mathcal{K} \\
& v_c \geq 0 \quad c \in \mathcal{C}^r.
\end{aligned}
$$

Using $\mathbf{p}^b$, $\mathbf{p}^a$, and $\mathbf{p}^n$ to denote the duals associated with the first three constraints and using q_k as the dual for the k^{th} convexity constraint, we define the dual ($\mathbf{ARM}_D$) as:

Model	Objective Value	Description
ESSND	z_{IP}	IP formulation of **ESSND**
ESSND$_{NLP}$	z_{NLP}	Master problem of **ESSND**
ESSND$_{LP}$	z_{LP}	LP relaxation of **ESSND**$_{NLP}$
ESSND$_{D}$	z_{D}	Dual of **ESSND**$_{LP}$
SUB$_{NLP}$	z_{SUB}	Subproblem in decomposition of **ESSND**
SUB$_{LP}$	$z_{SUB_{LP}}$	LP relaxation of **SUB**$_{NLP}$
ARM	z_{ARM}	Implementation of **ARM** with composite set $\mathcal{C}^r$
ARM$_{LP}$	$z_{ARM_{LP}}$	LP relaxation of **ARM**
ARM$_{D}$	z_{ARM_D}	Dual of **ARM**$_{LP}$

Table 6.1: Summary of models and notation

$$z^*_{ARM_D} = \max\ (\mathbf{p}^a)'\,\mathbf{a} + (\mathbf{p}^n)'\,\mathbf{n} + \sum_{k\in\mathcal{K}} q_k$$

$$\text{subject to} \quad \left(\mathbf{p}^b\right)'\mathbf{B}\mathbf{y}_c + (\mathbf{p}^a)'\,\mathbf{A}\mathbf{y}_c + (\mathbf{p}^n)'\,\mathbf{N}\mathbf{y}_c + \sum_{k\in\mathcal{K}} \delta_c^k q_k \le \mathbf{d}'\mathbf{y}_c \quad c \in \mathcal{C}^r$$

$$p_h^a \le 0 \quad h \in H$$

$$p_f^n \le 0 \quad f \in F$$

$$\mathbf{p}^b, q \text{ unrestr.}$$

We noted earlier the similarity between the master problem **ESSND**$_{NLP}$ and **ARM**. Although our practical implementation of **ARM** is solved over a reduced set of composites, $\mathcal{C}^r$, we can use that solution to derive bounds on **ESSND**$_{NLP}$ and, hence, **ESSND**. For clarity, we summarize the notation for models and their objective values in Table 6.1.

The dual (**ESSND**$_D$) is given by:

$$z^*_D = \max\ (\boldsymbol{\pi}^a)'\,\mathbf{a} + (\boldsymbol{\pi}^n)'\,\mathbf{n} + \sigma$$

$$\text{subject to} \quad (\boldsymbol{\pi}^a)'\,\mathbf{A}\mathbf{y}_c + \left(\boldsymbol{\pi}^b\right)'\mathbf{B}\mathbf{y}_c + (\boldsymbol{\pi}^n)'\,\mathbf{N}\mathbf{y}_c + \sigma \le \mathbf{d}'\mathbf{y}_c \quad c \in \mathcal{C}^*$$

$$\pi_h^a \le 0 \quad h \in H$$

$$\pi_f^n \le 0 \quad f \in F$$

$$\boldsymbol{\pi}^b, \sigma \text{ unrestr.}$$

Let $(\widehat{\mathbf{p}}, \widehat{\mathbf{q}})$ denote the optimal dual solution to **ARM**$_D$ found when solving **ARM**$_{LP}$. Letting

$\widehat{\sigma} = \sum_{k \in K} \widehat{q}_k$, we use $(\widehat{\mathbf{p}}, \widehat{\sigma})$ in the objective function of $\mathbf{SUB}_{NLP}$ and solve its LP relaxation to obtain the optimal value $z^*_{SUB_{LP}}$.

Lemma 33 *The solution $(\widehat{\mathbf{p}},\ z^*_{SUB_{LP}})$ is feasible for* $\boldsymbol{ESSND}_D$.

Proof. Solving the LP relaxation of $\mathbf{SUB}_{NLP}$ yields

$$\begin{aligned} z^*_{SUB_{LP}} &\leq z^*_{SUB} \\ &= \min_{c \in \mathcal{C}^*} \left[\mathbf{d}' - (\widehat{\mathbf{p}}^a)' \mathbf{A} - \left(\widehat{\mathbf{p}}^b\right)' \mathbf{B} - (\widehat{\mathbf{p}}^n)' \mathbf{N} \right] \mathbf{y}_c. \end{aligned}$$

Then $(\widehat{\mathbf{p}}^a)' \mathbf{A}\mathbf{y}_c + \left(\widehat{\mathbf{p}}^b\right)' \mathbf{B}\mathbf{y}_c + (\widehat{\mathbf{p}}^n)' \mathbf{N}\mathbf{y}_c + z^*_{SUB_{LP}} \leq \mathbf{d}'\mathbf{y}_c$ for all $c \in \mathcal{C}^*$. Thus, $(\widehat{\mathbf{p}},\ z^*_{SUB_{LP}})$ is dual feasible for **ESSND**. ■

This leads to the following lower bound:

Theorem 34 *A lower bound on the optimal cost of* ***ESSND*** *is given by* $z^*_{ARM_{LP}} + z^*_{SUB_{LP}} - \sum_{k \in \mathcal{K}} \widehat{q}_k$.

Proof. By weak duality and Lemma 33,

$$\begin{aligned} z^*_{IP} &= z^*_{NLP} \\ &\geq z^*_{LP} \\ &\geq (\widehat{\mathbf{p}}^a)' \mathbf{a} + (\widehat{\mathbf{p}}^n)' \mathbf{n} + z^*_{SUB_{LP}} \\ &= (\widehat{\mathbf{p}}^a)' \mathbf{a} + (\widehat{\mathbf{p}}^n)' \mathbf{n} + z^*_{SUB_{LP}} + \widehat{\sigma} - \widehat{\sigma} \\ &= z^*_{ARM_{LP}} + z^*_{SUB_{LP}} - \sum_{k \in \mathcal{K}} \widehat{q}_k. \end{aligned}$$

■

Thus, by solving two LP relaxations, we readily derive a lower bound on the optimal cost of **ESSND**. The first LP relaxation is that of the *implemented* version of **ARM** (which uses a subset of composite variables). The second is that of $\mathbf{SUB}_{LP}$ (the LP relaxation of the **ESSND** network loading subproblem). If we are able to find the *optimal integer* solution for $\mathbf{SUB}_{NLP}$ (with cost $z^*_{\mathbf{SUB}}$), Lemma 33 holds for the optimal integer solution to $\mathbf{SUB}_{NLP}$ and we can improve the overall bound to **ESSND** by the following corollary:

Corollary 35 *A lower bound on the optimal cost of the* ***ESSND*** *is given by* $z^*_{ARM_{LP}} + z^*_{SUB} - \sum_{k\in\mathcal{K}} \widehat{q}_k$.

Using an Integral ARM Solution

There is a second bound we can similarly produce when we have a *feasible integer* solution to **ARM**. Without meaningful dual information associated with this solution, we will construct a restricted master problem for **ESSND** and generate the dual values that we need to derive the new bound.

From a feasible integer **ARM** solution, specified by $\widetilde{\mathbf{v}}$ and having cost $\widetilde{z}_{ARM}$, we construct a *complete network solution* as follows:

$$\widetilde{\mathbf{y}} = \sum_{c\in C^r} \mathbf{y}_c \widetilde{v}_c.$$

Note that $\widetilde{\mathbf{y}}$ is a *feasible* solution to subproblem $\mathbf{SUB}_{NLP}$ because all gateway-hub demands are covered.

Let $\mathbf{ESSND}_{NLP'}$ denote the restricted master problem constructed with a single column corresponding to the complete network solution, $\widetilde{\mathbf{y}}$. The LP relaxation of this restricted master problem, which we call $\mathbf{ESSND}_{LP'}$, is given by:

$$\min \; \left[\mathbf{d}'\widetilde{\mathbf{y}}\right] v$$

$$\begin{aligned}
\text{subject to} \quad & [\mathbf{B}\widetilde{\mathbf{y}}]\, v = 0 \\
& [\mathbf{A}\widetilde{\mathbf{y}}]\, v \leq \mathbf{a} \\
& [\mathbf{N}\widetilde{\mathbf{y}}]\, v \leq \mathbf{n} \\
& v = 1 \\
& v \geq 0.
\end{aligned}$$

This clearly has only one feasible solution with cost $z^*_{LP'} = \mathbf{d}'\widetilde{\mathbf{y}} = \widetilde{z}_{ARM}$. Furthermore, the solution has integral aircraft routes, so $z^*_{NLP'} = z^*_{LP'}$.

The dual, denoted by $\mathbf{ESSND}_{D'}$, is given by:

$$\max\ (\boldsymbol{\pi}^a)'\,\mathbf{a} + (\boldsymbol{\pi}^n)'\,\mathbf{n} + \sigma$$

$$\begin{aligned} \text{subject to} \quad & (\boldsymbol{\pi}^a)'\,\mathbf{A}\widetilde{\mathbf{y}} + \left(\boldsymbol{\pi}^b\right)'\mathbf{B}\widetilde{\mathbf{y}} + (\boldsymbol{\pi}^n)'\,\mathbf{N}\widetilde{\mathbf{y}} + \sigma \leq \mathbf{d}'\widetilde{\mathbf{y}} \\ & \pi_h^a \leq 0 \quad h \in H \\ & \pi_f^n \leq 0 \quad f \in F \\ & \boldsymbol{\pi}^b, \sigma \text{ unrestr.} \end{aligned}$$

We solve $\mathbf{ESSND}_{D'}$ to obtain a set of dual variables $(\widetilde{\boldsymbol{\pi}}, \widetilde{\sigma})$ with optimal cost equal to $(\widetilde{\boldsymbol{\pi}}^a)'\,\mathbf{a} + (\widetilde{\boldsymbol{\pi}}^n)'\,\mathbf{n} + \widetilde{\sigma} = \mathbf{d}'\widetilde{\mathbf{y}}$ (by strong duality).

We use these duals in the $\mathbf{SUB}_{LP}$ objective function (denoting the instance by $\mathbf{SUB}_{\widetilde{LP}}$) and solve to obtain the optimal value $z^*_{SUB_{\widetilde{LP}}}$. Thus, by generating a feasible integer solution to **ARM** (with cost $\widetilde{z}_{ARM}$) and by solving two subsequent linear programs, we obtain the following lower bound on the optimal cost of **ESSND** (z^*_{IP}):

Theorem 36 *A lower bound on the optimal cost of* ***ESSND*** *is given by* $\widetilde{z}_{ARM} + z^*_{SUB_{\widetilde{LP}}} - \widetilde{\sigma}$.

Proof. From Lemma 33, $(\widetilde{\boldsymbol{\pi}},\ z^*_{SUB_{\widetilde{LP}}})$ is feasible for $\mathbf{ESSND}_D$. Then from weak duality

$$\begin{aligned} z^*_{IP} &\geq (\widetilde{\boldsymbol{\pi}}^a)'\,\mathbf{a} + (\widetilde{\boldsymbol{\pi}}^n)'\,\mathbf{n} + z^*_{SUB_{\widetilde{LP}}} \\ &= \left[(\widetilde{\boldsymbol{\pi}}^a)'\,\mathbf{a} + (\widetilde{\boldsymbol{\pi}}^n)'\,\mathbf{n} + \widetilde{\sigma}\right] + z^*_{SUB_{\widetilde{LP}}} - \widetilde{\sigma} \\ &= \widetilde{z}_{ARM} + z^*_{SUB_{\widetilde{LP}}} - \widetilde{\sigma}. \end{aligned}$$

■

Regardless of whether the complete network solution, $\widetilde{\mathbf{y}}$, is an extreme point of the $\mathbf{SUB}_{NLP}$ polyhedron, this bound still applies. In the decomposition algorithm, the presence of a non-extreme point in the restricted master problem has no effect on the generation of extreme points in subsequent iterations.

To summarize, the bounds based on integral solutions to **ARM** are derived similarly to those based on solutions to the **ARM** LP relaxation. The major difference is in the absence of dual information associated with the integral **ARM** solution. We obtain dual values by creating a trivial restricted master problem, and solving it to get its duals. These duals are used in exactly the same manner as the **ARM** duals were used in establishing the first bound.

In general, we cannot establish whether these bounds are tighter than those given by the **ESSND** LP relaxation (which are typically weak). Nonetheless, the new bounds are readily available at any stage in the Dantzig-Wolfe decomposition algorithm and might provide a better bound on the optimal **ESSND** solution.

6.1.6 ESSND Improvement Procedure Based on Decomposition

The decomposition provides a means for improving the restricted **ARM** solution. Assuming that commodities are specified by gateway-hub pairs, we create the restricted set of composites, $\mathcal{C}^r$, using the rules and routines presented in Chapter 3. If we remove the operational assumption regarding the limited use of double-leg routes, there are composites not included in $\mathcal{C}^r$ that might lead to a better solution. We can systematically examine these additional composites, perhaps by considering composites with limited interactions between aircraft routes. We then explicitly price-out these composites. Routes with negative reduced cost would be added to the restricted master problem until no such composites are found. We then branch on aircraft routes (recall that (6.25) enforce the integrality of aircraft routes), continuing to price-out composites at nodes in the branch-and-bound tree (i.e., Branch-and-Price). After finding the optimal (or improved) integral solution, we augment the set of composites by increasing the complexity of interactions between aircraft routes and continue the Branch-and-Price algorithm in search of an improved solution.

In this framework, we may relax the assumption about specifying commodities by gateway-hub pairs and, instead, specify them by gateway-gateway pairs. The initial **ARM** solution (found over $\mathcal{C}^r$) is certainly feasible with respect to the gateway-gateway commodities. We proceed with Branch-and-Price and the subproblem defined by (6.26)-(6.30) *plus* the hub sort constraints, which were removed under the assumption that commodities are specified by gateway-hub pairs. In other words, dual information found within the Dantzig-Wolfe framework can be used to help identify changes to gateway-hub assignments that will help identify improvements to the current **ARM** solution.

6.2 Dual Interpretation of ARM

In Chapter 5, we presented a family of models that began with **ESSND** and evolved into a routes-only model (**RO**) and, ultimately, into **ARM**. The transition from **RO** to **ARM** involved creating composite columns. The new columns are combinations of columns in **RO** with the capacity-demand coefficients of the new column rounded down (to 1 for coefficients greater than or equal to 1 and to 0 for coefficients less than 1). The idea of combining pieces of a linear program and rounding coefficients is a familiar idea. Cutting planes (due to Chvátal and Gomory and described in Nemhauser and Wolsey [64] and Cook et al. [25]) are the same type of operations but involve combinations of *rows*, not columns, to strengthen the LP relaxation. In this chapter, we explore the relationship between column operations used to create composite variables and row operations to create Chvátal-Gomory cuts in the dual of **RO**.

Here, we re-state the notation and formulation for the routes-only model, **RO**. The set $\mathcal{K}$ consists of gateway-hub commodities and is split into two disjoint sets $\mathcal{K}_P$ and $\mathcal{K}_D$ for the pickup and delivery sides, respectively. E is the set of extreme routes constructed from the aircraft routes and $\widehat{u}^e_{gh}$ is the *available capacity* provide by extreme route e for commodity (g, h). The matrices $\widehat{\mathbf{B}}$, $\widehat{\mathbf{A}}$, and $\widehat{\mathbf{N}}$ are the extreme route coefficients for the balance, landing, and plane count constraints. The indicator δ^{fr}_e maps each extreme route to aircraft route (f, r) by setting $\delta^{fr}_e = 1$. Finally, the decision variables are w_e, the number of times extreme route e is selected. The selection of each w_e is not required to be integral; rather, the aircraft routes built by the extreme routes must be integral.

The formulation for the routes-only model (**RO**) is given by:

$$\min \widehat{\mathbf{d}}'\mathbf{w} \tag{6.32}$$

$$\text{subject to} \quad \sum_{e \in E_P} \widehat{u}^e_{gh} w_e \geq b^{gh}_P \quad (g,h) \in \mathcal{K}_P \tag{6.33}$$

$$\sum_{e \in E_D} \widehat{u}^e_{gh} w_e \geq b^{gh}_D \quad (g,h) \in \mathcal{K}_D \tag{6.34}$$

$$\widehat{\mathbf{B}}\mathbf{w} = \mathbf{0} \tag{6.35}$$

$$\widehat{\mathbf{A}}\mathbf{w} \leq \mathbf{a} \tag{6.36}$$

$$\widehat{\mathbf{N}}\mathbf{w} \leq \mathbf{n} \tag{6.37}$$

$$\sum_{e \in E} \delta_e^{fr} w_e \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F. \tag{6.38}$$

The columns of **RO** correspond to the extreme routes of the aircraft routes, y_r^f, $r \in R^f$, $f \in F$. We consider the combinations of columns such that the resulting aircraft routes are integral and such that some subset of commodities are covered by these routes. Denote this combination by the index c. As shown in Chapter 5, the resulting coefficients corresponding to the aircraft balance constraints (6.21), the landing capacity constraints (6.22), and the plane count constraints (6.23) are integral.

The coefficients we examine more closely are those for the capacity-demand constraints. Let $\overline{\mathbf{u}}^c$ denote the vector of available capacities that result from the combination of extreme routes. Any component $\overline{u}_{gh}^c \geq b_P^{gh}$ indicates that capacity made available among the aircraft routes in the combination exceeds the demand to be moved. If the inequality is satisfied strictly, we round this coefficient down to b_P^{gh}. If we scale each capacity-demand constraint by its demand (right-hand-side), the coefficients represent the fraction of demand that is covered by each composite. When the fraction exceeds 1, we can reduce the coefficient to 1 as excess capacity does no good. When it is less than 1, we reduce it to zero.

Let's explore this with the **RO** formulation. We work with the LP relaxation of **RO** with the capacity-demand constraints scaled by their demands. The scaled capacity-demand coefficients, using the pickup routes for exposition, are defined by $\widehat{u}_{gh}^{*e} = \widehat{u}_{gh}^e / b_P^{gh}$ for all $e \in E_P$ and $(g,h) \in \mathcal{K}_P$. The formulation, denoted by $\mathbf{RO}_{LP}$, is:

$$\min \widehat{\mathbf{d}}'\mathbf{w} \tag{6.39}$$

$$\text{subject to} \quad \sum_{e \in E_P} \widehat{u}_{gh}^{*e} w_e \geq 1 \quad (g,h) \in \mathcal{K}_P \tag{6.40}$$

$$\sum_{e \in E_D} \widehat{u}_{gh}^{*e} w_e \geq 1 \quad (g,h) \in \mathcal{K}_D \tag{6.41}$$

$$\widehat{\mathbf{B}}\mathbf{w} = \mathbf{0} \tag{6.42}$$

$$\widehat{\mathbf{A}}\mathbf{w} \leq \mathbf{a} \tag{6.43}$$

$$\widehat{\mathbf{N}}\mathbf{w} \leq \mathbf{n} \tag{6.44}$$

$$w_e \geq 0 \quad r \in R^f,\ f \in F. \tag{6.45}$$

Denote the nonnegative dual variables associate with the covering constraints (6.40) and (6.41) to be $\boldsymbol{\mu}^P$ and $\boldsymbol{\mu}^D$, respectively (or $\boldsymbol{\mu}$ for the combination of the two). The dual variables with the remaining constraints are denoted by $\boldsymbol{\pi}^b$, which are unrestricted in sign; $\boldsymbol{\pi}^a$, which are nonpositive; and $\boldsymbol{\pi}^n$, which are also nonpositive. $\widehat{\mathbf{U}}_P$ and $\widehat{\mathbf{U}}_D$ denote the matrices of scaled available capacities ($\widehat{u}^{*e}_{gh}$) for the pickup side and delivery side, respectively. The dual problem, $\mathbf{RO}_D$, is:

$$\max\ \mathbf{e}'\boldsymbol{\mu} + \mathbf{a}'\boldsymbol{\pi}^a + \mathbf{n}'\boldsymbol{\pi}^n \tag{6.46}$$

$$\begin{aligned} \text{subject to} \quad & \widehat{\mathbf{U}}'_P\boldsymbol{\mu}^P + \widehat{\mathbf{U}}'_D\boldsymbol{\mu}^D + \widehat{\mathbf{B}}'\boldsymbol{\pi}^b + \widehat{\mathbf{A}}'\boldsymbol{\pi}^a + \widehat{\mathbf{N}}'\boldsymbol{\pi}^n \leq \mathbf{d} \\ & \mu^P_{gh},\ \mu^D_{gh} \geq 0 \quad (g,h) \in \mathcal{K}_D \\ & \pi^a_h \leq 0 \quad h \in H \\ & \pi^n_f \leq 0 \quad f \in F \\ & \boldsymbol{\pi}^b \text{ unrestr.} \end{aligned}$$

Without loss of generality, consider two rows, denoted by e_1 and e_2, corresponding to extreme routes on the pickup side. Let $\widehat{\mathbf{B}}_e$ denote the e^{th} column of the constraint matrix $\widehat{\mathbf{B}}$. The dual constraint associated with e_1 is given by

$$\sum_{(g,h)\in\mathcal{K}_P} \widehat{u}^{*e_1}_{gh}\mu^P_{gh} + \widehat{\mathbf{B}}'_{e_1}\boldsymbol{\pi}^b + \widehat{\mathbf{A}}'_{e_1}\boldsymbol{\pi}^a + \widehat{\mathbf{N}}'_{e_1}\boldsymbol{\pi}^n \leq d_{e_1}$$

and the constraint associated with e_2 is given by

$$\sum_{(g,h)\in\mathcal{K}_P} \widehat{u}^{*e_2}_{gh}\mu^P_{gh} + \widehat{\mathbf{B}}'_{e_2}\boldsymbol{\pi}^b + \widehat{\mathbf{A}}'_{e_2}\boldsymbol{\pi}^a + \widehat{\mathbf{N}}'_{e_2}\boldsymbol{\pi}^n \leq d_{e_2}.$$

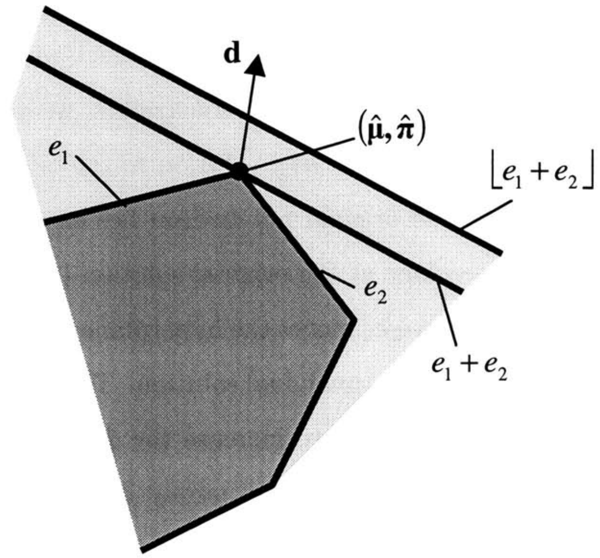

Figure 6-1: The effect of composite variables in the dual space

Adding the two rows yields

$$\sum_{(g,h)\in\mathcal{K}_P} \left(\widehat{u}_{gh}^{*e_1} + \widehat{u}_{gh}^{*e_2}\right)\mu_{gh}^P + \left(\widehat{\mathbf{B}}'_{e_1} + \widehat{\mathbf{B}}'_{e_2}\right)\boldsymbol{\pi}^b + \left(\widehat{\mathbf{A}}'_{e_1} + \widehat{\mathbf{A}}'_{e_2}\right)\boldsymbol{\pi}^a + \left(\widehat{\mathbf{N}}'_{e_1} + \widehat{\mathbf{N}}'_{e_2}\right)\boldsymbol{\pi}^n \leq d_{e_1} + d_{e_2}. \tag{6.47}$$

Recall the process of rounding described in Chapter 5. After combining extreme route columns, we round down the coefficients. The elements from $\widehat{\mathbf{B}}$, $\widehat{\mathbf{A}}$, and $\widehat{\mathbf{N}}$ are integral and are not affected by the rounding. The capacity-demand coefficients in $\widehat{\mathbf{U}}$ are not restricted to integer values so rounding *can* affect the sum $\widehat{u}_{gh}^{*e_1} + \widehat{u}_{gh}^{*e_2}$. For all $(g,h) \in \mathcal{K}$ we are rounding the coefficient down to 1 if $\widehat{u}_{gh}^{*e_1} + \widehat{u}_{gh}^{*e_2} > 1$ and rounding down to 0 if $\widehat{u}_{gh}^{*e_1} + \widehat{u}_{gh}^{*e_2} < 1$. With the dual variables $\boldsymbol{\mu}$ nonnegative, this rounding adds slack to (6.47). Any dual solution $(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\pi}})$ that satisfies (6.47) with equality will satisfy the rounded version of (6.47), with strict *inequality*. With slack, there is a direction we can move and still ensure dual feasibility. This movement can be represented by a positive change in at least one of the dual variables. Because the objective function coefficient for each element of $\boldsymbol{\mu}$ is unity, the feasible direction is also a direction of cost increase. Thus, the increase in slack created by reducing the capacity demand coefficients strictly *increases* the dual objective value.

This phenomenon is depicted in Figure 6-1. The dark region represents the original dual polyhedron before the extreme routes have been combined and rounded. The optimal dual **RO** solution is $(\widehat{\mu}, \widehat{\pi})$. When we sum the two constraints, we get a new hyperplane (denoted by $e_1 + e_2$) that passes through $(\widehat{\mu}, \widehat{\pi})$. Any solution to the original problem satisfies the new constraint and, consequently, the original polyhedron lies entirely in a half-space determined by $e_1 + e_2$, touching the hyperplane at the original solution $(\widehat{\mu}, \widehat{\pi})$.

Rounding the coefficients in $e_1 + e_2$ moves the hyperplane away from the original polyhedron. This rounding has *no* effect on the optimal dual solution. This constraint is weaker (in the dual) than the original constraints. In order to increase the dual objective value, we must *remove* the original constraints and state the problem purely in terms of the new constraint. Only then will we see an increase in the dual objective function and, hence, in the primal objective function. This accounts for the tightening of the lower bounds when we solve the **ARM** LP relaxation versus the **RO** LP relaxation, but only when each **ARM** variable is a composite cover.

The mechanics of this approach (as viewed in the dual) are similar to the mechanics of cutting plane techniques of Gomory and Chvátal. The key difference is that cutting plane techniques are designed to remove fractionality and tighten the approximation that the LP relaxation provides for the optimal integer solution. In the dual of composite variable formulations, the "combine and round" technique weakens the dual, giving the primal a tighter bound on the optimal integer solution.

6.3 Composite Variable Formulation for the Pure Fixed Charge Transportation Problem

Having interpreted **ARM** in the general settings of Dantzig-Wolfe decomposition and duality, we now explore the general idea of composite variable formulations on a different class of problem. In solving the **ESSND** problem, the composite variable approach relies on our ability to separate the problem by *commodity*. In this section, we consider a single commodity network design problem where a similar decomposition strategy is used but applied to give coverage to *locations*, not commodities. The effect is the same: we solve subproblems that generate extreme points (composite covers) that either originate from or terminate at a given node in the network. The master problem of this decomposition is a simple set partitioning problem with side constraints and the subproblems are small 0-1 knapsack problems.

We consider an uncapacitated transportation problem in which fixed costs are incurred if an arc contains positive flow. We refer to this problem as "pure" since the fixed charge is the only cost incurred. We are given a bipartite network $G = (N, A)$ where one partition, denoted by N_O, consists of source nodes and the other partition, denoted by N_D, consists of sink nodes. That is, $b_i > 0$ for $i \in N_O$, $b_j < 0$ for $j \in N_D$, where $N = N_O \cup N_D$ and $\sum_{i \in N_O} b_i = -\sum_{j \in N_D} b_j$. Each arc in the network, if utilized, incurs a fixed cost, d_{ij}, and has unlimited capacity.

We take advantage of an "effective capacity" for each arc, recognizing that the amount of flow on each arc is a function of the demands of the two nodes it connects. We define the effective capacity of arc (i, j) to be $\rho_{ij} = \min(b_i, -b_j)$. We use this in lieu of "big-M" in the forcing constraints to provide a tighter formulation.

Our decisions are whether to open arc (i, j), represented by the binary variable y_{ij}, and how much flow to place on arc (i, j), represented by the continuous variable x_{ij}. The Pure Fixed Charge Transportation Problem (**PFCTP**) is given by:

$$\min \sum_{(i,j) \in A} d_{ij} y_{ij}$$

$$\text{subject to} \quad \sum_{j:(i,j) \in A} x_{ij} = b_i \quad i \in N_O \tag{6.48}$$

$$\sum_{i:(i,j) \in A} x_{ij} = b_j \quad j \in N_D \tag{6.49}$$

$$x_{ij} \leq \rho_{ij} y_{ij} \tag{6.50}$$

$$x_{ij} \geq 0 \quad (i, j) \in A$$

$$y_{ij} \in \{0, 1\} \quad (i, j) \in A.$$

Our composite variable formulation strategy follows two steps. First, we create a 0-1 integer program that is equivalent to **PFCTP**. This new model is analogous to the **RO** model from Chapter 5. We then group these decision variables into *composites*, where each composite covers the demand at a specific node, and construct a composite variable formulation.

The idea of transforming this problem into a 0-1 integer programming problem is not unique. Göthe-Lundgren and Larsson [38] map feasible solutions of the **PFCTP** to a $\{0,1\}^{|N_O|\times|N_D|}$ hypercube and present a formulation with exponentially many constraints. They introduce a separation algorithm for identifying violated constraints. The overall result is a solution methodology that quickly generates near-optimal solutions for the PFCTP. The approach we take is similar in the sense that the variables in the master problem are $\{0,1\}$. The difference is that instead of generating violated inequalities, we generate composite columns.

For the first step in the composite variable formulation strategy, combining constraints (6.48) and (6.50) allows us to remove the package flow variables. That is, for any $i \in N_O$, we have

$$b_i = \sum_{j:(i,j)\in A} x_{ij} \leq \sum_{j:(i,j)\in A} \rho_{ij} y_{ij}. \tag{6.51}$$

Similarly on the sink side, we use (6.49) and (6.50) so that for each $j \in N_D$, we have

$$b_j = \sum_{i:(i,j)\in A} x_{ij} \leq \sum_{i:(i,j)\in A} \rho_{ij} y_{ij}. \tag{6.52}$$

Selecting any set of arcs such that $\sum_{j:(i,j)\in A} \rho_{ij} y_{ij} \geq b_i$ implies the existence of a feasible package flow out of node $i \in N_O$ (and similarly for any node $j \in N_D$).

Next, we make a copy of the decision variables and associate one copy with the source side of the network and one copy with the sink side (similar to Lagrangean Decomposition described in Guignard and Kim [40]). Denote the two vectors of decision variables by $\mathbf{y}^o$ and $\mathbf{y}^d$. We re-write **PFCTP** in the following "routes-only" form:

$$\begin{aligned} &\min \sum_{(i,j)\in A} d_{ij} y^o_{ij} \\ &\text{subject to} \quad y^o_{ij} - y^d_{ij} = 0 \quad (i,j) \in A \\ &\sum_{j:(i,j)\in A} \rho_{ij} y^o_{ij} \geq b_i \quad i \in N_O \end{aligned} \tag{6.53}$$

$$\sum_{i:(i,j)\in A} \rho_{ij} y^d_{ij} \geq b_j \quad j \in N_D \tag{6.54}$$

$$y^o_{ij},\ y^d_{ij} \in \{0,1\} \quad (i,j) \in A.$$

Let C^i be the set of extreme points of the integer (0-1) hull of constraint (6.53) corresponding to each source node $i \in N_O$. We denote the extreme points as $\mathbf{y}^i_c$, $c \in C^i$. We define C^j and $\mathbf{y}^j_c$ similarly for the sink nodes $j \in N_D$. These extreme points are simply composites that cover either the source demand or the sink demand.

$$\min \sum_{i\in N_O} \sum_{c\in C^i} \left[\mathbf{d}'\mathbf{y}^i_c\right] v_c \tag{6.55}$$

$$\text{subject to} \quad \sum_{i\in N_O} \sum_{c\in C^i} \mathbf{y}^i_c v_c - \sum_{j\in N_D} \sum_{c\in C^j} \mathbf{y}^j_c v_c = \mathbf{0} \tag{6.56}$$

$$\sum_{c\in C^i} v_c = 1 \quad i \in N_O \tag{6.57}$$

$$\sum_{c\in C^j} v_c = 1 \quad j \in N_D \tag{6.58}$$

$$v_c \in \{0,1\} \quad c \in C. \tag{6.59}$$

Each variable represents a group of local design variables (i.e., groups of "open" arcs rooted at a common node) with adequate capacity to carry all demand from that node. In the context of a column generation algorithm, the subproblems provide the mechanism to implicitly generate composite variables with negative reduced cost, which are subsequently added as columns in the restricted master problem.

We formulate and solve the subproblems as follows. Define $\boldsymbol{\sigma}^o$ ($\ni \sigma^o_i$, $i \in N_O$) to be the dual variables associated with constraints (6.57), $\boldsymbol{\sigma}^d$ ($\ni \sigma^d_i$, $i \in N_D$) to be the dual variables associated with constraints (6.58), and $\boldsymbol{\pi}$ be the vector of dual variables for constraints (6.56). We define $A(i)$ to be the set of arcs incident to node $i \in N$ and, for subproblem i, we only allow the selection of arcs in $A(i)$. Using $\boldsymbol{\rho}$ to denote the vector of effective arc capacities, the subproblem for source $i \in N_O$ is given by:

$$\min \ [\mathbf{h} - \boldsymbol{\pi}]' \, \mathbf{y} \tag{6.60}$$

$$\text{subject to} \quad \boldsymbol{\rho}'\mathbf{y} \geq b_i$$

$$y_{ij} \in \{0,1\} \quad (i,j) \in A(i)$$

and the subproblem for sink $j \in N_D$ is given by:

$$\min \ \boldsymbol{\pi}'\mathbf{y}$$

$$\text{subject to} \quad \boldsymbol{\rho}'\mathbf{y} \geq b_j$$

$$y_{ij} \in \{0,1\} \quad (i,j) \in A(j).$$

If the optimal solution to source subproblem i has cost less than σ_i^o, then the extreme point has negative reduced cost and is added to the restricted master problem. Similarly, if the optimal solution to sink subproblem j has cost less than σ_j^d, the extreme point is added to the restricted master.

We turn now to the solution of the subproblems, which we accomplish by transforming each subproblem to a 0-1 knapsack problem. Let $\mathbf{e}$ denote the vector of ones with length equal to the length of $\mathbf{y}$. Consider the transformation $\mathbf{z} = \mathbf{e} - \mathbf{y}$. With some manipulation, the j^{th} subproblem is written in terms of $\mathbf{z}$ as the following 0-1 knapsack problem:

$$\max \ \boldsymbol{\pi}'\mathbf{z} - \boldsymbol{\pi}'\mathbf{e}$$

$$\text{subject to} \quad \boldsymbol{\rho}'\mathbf{z} \leq \boldsymbol{\rho}'\mathbf{e} - b_j$$

$$z_{ij} \in \{0,1\} \quad (i,j) \in A(j).$$

The terms $\boldsymbol{\pi}'\mathbf{e}$ and $\boldsymbol{\rho}'\mathbf{e} - b_j$ are constants. If a feasible solution to (6.60) exists, then it must be the case that $\boldsymbol{\rho}'\mathbf{e} - b_j > 0$ and the knapsack problem is well-defined. Thus, by solving the 0-1 knapsack problem, we can easily transform its solution to a solution of the original subproblem.

Example 37 *We consider the network shown in Figure 6-2. At each node we show the demand and on each arc we show the fixed charge associated with "opening the arc." The capacity of each arc is unlimited. We determine the values of* $\boldsymbol{\rho}$ *to be:*

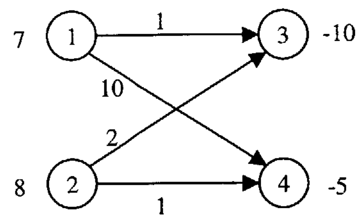

Figure 6-2: Four node fixed charge transportation network

$$\begin{aligned}\rho_{13} &= \min(7,10) = 7\\ \rho_{14} &= \min(7,5) = 5\\ \rho_{23} &= \min(8,10) = 8\\ \rho_{24} &= \min(8,5) = 5\end{aligned}$$

*We formulate **PFCTP** as follows:*

$$\min\ y_{13} + 10y_{14} + 2y_{23} + y_{24}$$

$$\begin{aligned}\textit{subject to}\quad x_{13} + x_{14} &= 7\\ x_{23} + x_{24} &= 8\\ x_{13} + x_{23} &= 10\\ x_{14} + x_{24} &= 5\\ x_{13} - 7y_{13} &\leq 0\\ x_{14} - 5y_{14} &\leq 0\\ x_{23} - 8y_{23} &\leq 0\\ x_{24} - 5y_{24} &\leq 0\\ x_{ij} &\geq 0,\ y_{ij} \in \{0,1\}\end{aligned}$$

The coefficients on the $\mathbf{y}$ *variables in the forcing constraints are* $\rho_{ij} = \min(b_i, -b_j)$. *The LP relaxation has cost 2.75 and the relaxed solution for the integer variables is* $y_{13} = 1.0$, $y_{23} =$

0.375, *and* $y_{24} = 1$. *The optimal integer solution has cost 4.00.*

For the composite variable formulation (i.e., the design-only master problem) we define the composite variables (with respect to the sources) a priori. On the source side, we define four composite covers:

$$\mathcal{C}_O = \{\{(1,3)\},\{(1,3),(1,4)\},\{(2,3)\},\{(2,3),(2,4)\}\}.$$

On the sink side, we define three composite covers:

$$\mathcal{C}_D = \{\{(1,3),(2,3)\},\{(1,4)\},\{(2,4)\}\}.$$

The composite variable formulation (i.e., the full master problem) for this example is

$$\min\ v^o_{13} + 11v^o_{13,14} + 2v^o_{23} + 3v^o_{23,24}$$

$$\begin{aligned}
\textit{subject to}\quad v^o_{13} + v^o_{13,14} - v^d_{13,23} &= 0\\
v^o_{13,14} - v^d_{14} &= 0\\
v^o_{23} + v^o_{23,24} - v^d_{13,23} &= 0\\
v^o_{23,24} - v^d_{24} &= 0\\
v^o_{13} + v^o_{13,14} &= 1\\
v^o_{23} + v^o_{23,24} &= 1\\
v^d_{13,23} &= 1\\
v^d_{14} + v^d_{24} &= 1\\
v^o_{ij},\ v^d_{ij} &\in \{0,1\} \quad (i,j) \in A.
\end{aligned}$$

Here, the cost of the LP relaxation yields the optimal integer solution with cost 4.0. The solution is $v^o_{13} = 1$, $v^o_{23,24} = 1$, $v^d_{13,23} = 1$, *and* $v^d_{24} = 1$.

6.4 Summary

We have presented the idea of composite variable formulations as an alternative Dantzig-Wolfe decomposition of the **ESSND** problem. Composite covers, which for **ESSND** are collections of aircraft routes on which a feasible flow exists for some set of commodities, are related to the extreme point solutions of subproblems in the decomposition. The master problem of this decomposition yields the Aircraft Routing Model (**ARM**). The primary benefit of this form of decomposition is the removal of forcing constraints from the master problem by embedding the flow information within the extreme points of the master problem.

For certain classes of problems, the subproblem may be separable, allowing easier generation of integer extreme points. In other problems, as in the case of **ARM**, we artificially create this separation. After generating a solution to **ARM** (by considering a subset of all possible composites), we can use the decomposition framework to pursue the optimal solution to **ESSND** or, at a minimum, an integer solution with improved cost. Finally, in spite of restricting the **ARM** solution to consider only gateway-hub commodities, the decomposition framework can examine a broader set of solutions by defining the commodities as in the original **ESSND** gateway-to-gateway demands.

We may use this decomposition and our solution to **ARM** (over the restricted set of composites) to derive lower bounds on the optimal cost of the original **ESSND** problem. These bounds provide insight into whether trying to improve the current **ARM** solution will yield significant improvement. One bound was based on the solution to the LP relaxation of **ARM** and the solution of the Dantzig-Wolfe subproblem using the **ARM** dual variables in the objective function. A second bound was based on the integer (feasible) solution to **ARM**, the solution of the *restricted* master problem dual, and the solution of the Dantzig-Wolfe subproblem, this time using the duals from the restricted master.

We also compare **ARM** to cutting plane methods applied to the dual of the routes-only model, **RO**. Combining the columns of **RO** and rounding the capacity-demand coefficients in the primal is equivalent to combining constraints and rounding coefficients in the dual. This is similar to the method of Chvátal-Gomory cuts, but the rounding actually "*weakens*" the dual formulation. This is consistent, however, with what we observe in the primal. The dual feasible region expands and yields a higher objective value. The corresponding primal solution also increases, thus providing a tighter bound on the optimal integer solution to **ESSND**.

Finally, we extend the idea of composite covers to the case where the separability of the subproblem is achieved by source and sink nodes in a bipartite transportation network. Through the Pure Cost Fixed Transportation Problem, we demonstrate on a single commodity network design problem the same ideas of creating composites/integer extreme points via subproblems and solving a design-only master problem to yield a solution to the original problem. A numerical example demonstrated the stronger bounds provided by the LP relaxation of the composite variable formulation versus the LP relaxation of the original **PFCTP** problem.

Chapter 7

Conclusions and Future Work

Composite variable formulations represent a new approach for solving network design problems. While this thesis centered on the Express Shipment Service Network Design (**ESSND**) problem, there is evidence suggesting a broad applicability of this method. In fact, we demonstrated an example of this approach on the Pure Fixed Charge Transportation Problem simply to illustrate how this may be done.

The composite variable approach consists of two major components. First, we represent flow decisions implicitly in the design variables, thus eliminating the need to model flows explicitly. Second, we combine the design variables (in our case, aircraft routes) into *composites,* with the important feature being that composites fully cover the demand of one or more commodities. The resulting model, a set covering formulation with side constraints, is more easily solved than traditional network design formulations in which flow and design decisions are modeled explicitly.

The major contributions of this thesis are the following:

- Developed a **robust solution methodology** for solving the Express Shipment Service Network Design (**ESSND**) problem. Standard polyhedral methods for network design and network loading problems are not effective on problem instances of realistic size. The composite variable formulation provides stronger bounds combined with the flexibility to handle practical constraints that further limit our ability to solve these problems using traditional formulations. The model has fast run times, making it a useful tool to support network planners.

- Demonstrated the **practical significance** of the composite variable approach on a carrier-specific instance of the **ESSND** problem. This instance, which is representative of many others, could not be solved otherwise. We demonstrated the potential to save hundreds of millions of dollars in the annual cost of owning and operating aircraft.

- Demonstrated the **theoretical foundation** for this method. We established the equivalence of this integer programming formulation with the mixed integer programming formulation **ESSND** and the fact that the composite variable formulation provides stronger bounds on the optimal integer solution.

- Demonstrated **how to generalize** the composite variable approach to a broader class of problems. We did this by relating composite variable formulations to Dantzig-Wolfe decomposition and relating the specific act of creating a composite to the cutting plane methods of Chvátal and Gomory.

This thesis represents a first step in modeling with composite variable formulations. In the course of developing this strategy and implementing the composite variable formulation for the UPS planning problem, future areas of research became evident. The broad areas include generalizing the composite variable method as well as enhancing the Aircraft Routing Model (**ARM**). Specific areas of further study with respect to **ARM** include:

- *Solving a combined Next-Day/Second-Day version of **ARM**.* The combined network includes both daytime and overnight operations. The core elements of the problem are the same. The difficulty lies in the fact that we must account for cargo that is spending more time in the system. Thus, instead of dealing with a 24-hour repeatable network, we must create a network that repeats weekly.

- *Methods to create, or suggest improvements to, the gateway-hub assignments.* Currently, we take these assignments as fixed and define the commodities according to gateway-hub pairs. Allowing the model to choose these assignments will provide additional opportunities to improve the solution.

- *Expand the set of composite variables over which we solve **ARM**.* Currently, we enforce an operational assumption that limits the composites we build by requiring double-leg routes to pick up at least one of its gateway-hub demands in full. Relaxing this assumption will increase the complexity of the composites, in the sense that greater interactions can occur between aircraft routes.

- *Column generation techniques.* As we increase the number of composites, the large number of columns in **ARM** will pose computational burdens in the form of excessive memory requirements and long running times. We should implement algorithms to price-out columns, either explicitly or implicitly, and we should build a branch-and-price framework to search for the optimal integer solution.

- *Explore the use of the routes-only formulation as more than a proof tool.* The first step would be to re-define the extreme routes as the extreme points of the feasible flow polyhedron, not the maximum flow polyhedron (see Chapter 6). This would allow us to select aircraft routes (as convex combinations of extreme routes) so that the available capacity more closely approximates the actual flow on the route.

In addition to changes specific to the express shipment problem, we suggest the following areas of research relate to the general application of composite variable formulations:

- *Implementing and performing computational testing on the Pure Fixed Charge Transportation Problem (**PFCTP**).* The **PFCTP** problem is a basic network design problem in which we are moving a single commodity from multiple sources to multiple sinks. A key aspect of the **PFCTP** composite variable formulation is the use of *location-based* covering constraints versus *commodity-based* covering. The **ESSND** formulation used the second form of covering. For more general problems, utilizing a mixture of approaches might be necessary as multiple commodities may be specified with multiple origins and multiple destinations. Additionally, we should extend the composite variable formulation to the capacitated version of the **PFCTP**.

- *Accounting for costs on flows.* In some cases (like the **PFCTP**), flow costs are zero. In many real-world problems (such as the one we considered in this thesis), the flow costs are orders of magnitude smaller than the cost of the design elements and we treat them as zero. When we need to account for flow costs, how we do so depends upon the type of problem we are solving. In certain cases, we can take advantage of the fact that the minimum cost flow within a given composite will still be optimal when that composite is chosen as part of the complete network solution. In other situations, this is not the case and a different approach would be required.

- *Cases when the design variables and flow paths do not have the same "scope."* In **ESSND**, the paths on which commodities are flown are the same as the paths to which aircraft are assigned. In **PFCTP**, these paths are simply arcs. In a more general setting, we need to consider situations where, for instance, flows occur on paths and capacity is installed on individual arcs.

- *Applying composite variable formulations to telecommunications problems.* Much of the work in network design has occurred in this application domain. We should consider both the bifurcated case (flow splitting is allowed) and the non-bifurcated case (no flow splitting is allowed). Initial work should be directed at the non-bifurcated case, which has similar assumptions to those we exploited (i.e., no ramp transfer) when establishing the equivalence of **ARM** and **ESSND**.

- *Combining the composite variable formulation with other solution strategies.* It is possible that the difficulty in solving a particular instance of a network design problem stems from some small portion of the network. We should explore methods for identifying those complex areas and use the composite variable approach to "handle" those portions of the network. We could then interweave the resulting subnetwork models with the remaining "easy" portion of the network, which may be solved using efficient special-purpose algorithms.

- *Application to other fixed-charge problems.* One logical place to start is with facility location problems. Initially, we might consider a version with no flow costs. This will allow us to isolate the techniques by which we can represent the opening of a node versus the opening of an arc. Many facility location problems, however, have significant flow cost so understanding how to model flow costs is imperative.

Appendix A

Glossary

AIRCRAFT ROUTE: Combination of an aircraft of a particular **fleet** type flying a particular **route**.

ARM: The Aircraft Routing Model.

BLOCK TIME: The time an aircraft is in service on a given route; equals flying time plus taxi time.

CYCLE COST: Fixed cost incurred each time a **leg** is flown.

DELIVERY ROUTE: A **route** from a **hub** to one (or two) **gateways**. A delivery route departs the hub not earlier than the **sort end time** and must arrive at each gateway by the gateway's **Late Delivery Time**.

DELIVERY SIDE: Refers to the entire operation of flying **delivery routes**.

DROP-OFF (DROP-ON) ROUTE: A double-leg **pickup (delivery) route** that delivers (picks up) packages to (from) its intermediate location, which can be either a **gateway** or a **hub**.

EARLY PICKUP TIME: The earliest time a plane may depart a **gateway** location on a **pickup route**.

ESSND: The Express Shipment Service Network Design Formulation.

FEEDER AIRCRAFT: Small aircraft that transport packages from remote locations to **gateway** locations. These planes are also available within the **next-day air network** to augment the available capacity.

FLEET: Refers to a group of aircraft of the same type.

GATEWAY: An airport at which packages are transferred from ground vehicles and **feeder aircraft** to the airplanes that fly the **pickup routes**. Gateways are also the points at which packages are transferred from airplanes that fly the **delivery routes** to ground vehicles and **feeder aircraft**

HUB: Airport at which packages are sorted. Hubs serve as the terminating location for **pickup routes** and the starting point for **delivery routes**.

LATE DELIVERY TIME: The latest time a plane may arrive at a **gateway** location on a **delivery route**.

LEG: An aircraft movement between successive **gateways** on a particular **route**.

NEXT-DAY AIR (NDA) NETWORK: The set of air routes operated to pickup and delivery packages that require overnight delivery.

O-D DEMAND: Set of packages moving from a common Origin to a common Destination.

HOURLY COST: The variable cost of operating an aircraft, applied to **block hours** flown, and specified by **fleet** type.

OWNERSHIP COST: The daily cost of owning an aircraft.

PICKUP ROUTE: A **route** from one (or two) **gateways** to a **hub**. A pickup route departs each gateway location not earlier than each gateway's **Earliest Pickup Time** and arrives at the hub not later than the hub's **sort start time**.

PICKUP SIDE: Refers to the entire operation of flying **pickup routes**.

RAMP TRANSFER: The act of transferring packages from one plane to another at an intermediate **gateway**. Ramp transfers occur only when the system has multiple **hubs**.

RO: The Routes-Only formulation.

ROUTE: An ordered set of **gateway** locations and a **hub**.

SECOND-DAY AIR (SDA) NETWORK: The set of air routes operated to pickup and delivery packages that require second day delivery. These operations occur during the daytime.

SORT: The main activity of the **hubs**. During the sort, packages are removed from **pickup routes** and systematically loaded onto **delivery routes**.

SORT START: The time at which the sort begins at a **hub**. This is also the latest time at which planes can arrive to the hub on their **pickup routes**.

SORT END: The time at which the sort stops at a **hub**. This is also the earliest time at which planes can depart the hub on their **delivery routes**.

TAXI TIME: For a given **leg**, this is the total time a plane spends traveling to the runway prior to taking off and traveling from the runway after landing.

TURN TIME: The time required for an airplane to stay on the ground (not including **taxi time**) between **legs**.

Sets:

$[S,T]$	Cutset including arcs (routes) originating in $S \subset N$ and terminating in $T \subset N$
A	Set of flight arcs in time-space network
$\mathcal{C}$	Set of composite variables
$\mathcal{C}^r$	Restricted set of composite variables in UPS implementation of **ARM**
$D(k)$	Destination of commodity $k \in \mathcal{K}$
E	Set of extreme routes
F	Set of aircraft (fleet) types (also set of facility types in general network loading)
G	Set of gateway locations
H	Set of hub locations
$\mathcal{K}$	Set of commodities (general)
$\mathcal{K}_c$	Set of commodities covered by composite $c \in \mathcal{C}$
K	Set of commodities (origin-destination)
K^s	Set of O-D commodities included in supercommodity $s \in S$
N	Set of nodes in time-space network
$O(k)$	Origin of commodity $k \in \mathcal{K}$
P^k	Set of paths from $O(k)$ to $D(k)$
P_c^{gh}	Paths between gateway $g \in G$ and hub $h \in H$ in composite $c \in \mathcal{C}$
$Q(g)$	Set of locations in the neighborhood of $g \in G$ used for quasi-balance
R	Set of routes (for aircraft)
R_P	Set of pickup routes
R_D	Set delivery routes
R^f	Routes that can be flown by fleet type $f \in F$
$R(\bar{i})$	Routes that originate at location $i \in G \cup H$
$R(\underline{i})$	Routes that terminate at location $i \in G \cup H$
$R(g,h)$	Routes that connect gateway $g \in G$ with hub $h \in H$
S	Set of supercommodities (O-D commodities rooted at a common origin)

Data:

$\mathbf{A}$	Landing (Arrival) constraint matrix for **ESSND**
$\widehat{\mathbf{A}}$	Landing (Arrival) constraint matrix for **RO**
$\overline{\mathbf{A}}$	Landing (Arrival) constraint matrix for **ARM**
$\mathbf{a}$	Landing capacities, $\ni a_h,\ h \in H$
$\mathbf{B}$	Aircraft balance constraint matrix for **ESSND**
$\widehat{\mathbf{B}}$	Aircraft balance constraint matrix for **RO**
$\overline{\mathbf{B}}$	Aircraft balance constraint matrix for **ARM**
b^k	Demand of O-D commodity $k \in K$
b^{gh}	Demand of gatway-hub commodity, subscripted appropriately by P for pickup and D for delivery
$\mathbf{b}^k$	$(\ni b_i^k,\ i \in G \cup H)$ Vector of node demands for commodity $k \in \mathcal{K}$
$\mathbf{b}^s$	$(\ni b_i^s,\ i \in G \cup H)$ Vector of node demands for supercommodity $s \in S$
$\mathbf{c}^k$	$(\ni c_{ij}^k,\ (i,j) \in A)$ Arc cost for commodity $k \in \mathcal{K}$ on arc $(i,j) \in A$
$\mathbf{c}^s$	$(\ni c_{ijk}^s,\ (i,j) \in A,\ k \in K^s)$ Commodity-specific arc costs for supercommodity $s \in S$
$D_{S,T}$	Aggregate demand across cutset $\{S,T\}$
$D_{S,T}^c$	Aggregate demand across cutset $\{S,T\}$ covered by composite $c \in \mathcal{C}$
d_r^f	Cost of flying aircraft route (f,r)
d_{ij}	Fixed cost of opening arc $(i,j) \in A$ in general network design problems
d_{ij}^f	Fixed cost of assigning one unit of facility type $f \in F$ in network loading problems
d_c	Cost of flying composite $c \in \mathcal{C}$
$\mathbf{e}$	$(\ni e_h,\ h \in H)$ Vector of hub sort capacities
$\mathcal{N}^k$	Node-arc incidence matrix for commodity $k \in \mathcal{K}$
$\mathbf{N}$	Plane count constraint matrix for **ESSND**
$\widehat{\mathbf{N}}$	Plane count constraint matrix for **RO**
$\overline{\mathbf{N}}$	Plane count constraint matrix for **ARM**
$\mathbf{n}$	$(\ni n_f,\ f \in F)$ Vector of fleet sizes
$\mathbf{U}$	Capacity matrix for forcing constraints in **ESSND**
$\widehat{\mathbf{U}}$	Capacity matrix for capacity-demand constraints in **RO**
u_{ij}	Capacity assigned to arc $(i,j) \in A$ in general network design problems

Data (cont'd):

u_{ij}^f Capacity of facility $f \in F$ assigned to arc $(i,j) \in A$ in network loading problems

$\widehat{u}_{gh}^e$ Available capacity for commodity (g,h) in extreme route $e \in E$

$\overline{u}_{gh}^c$ Available capacity for commodity (g,h) in composite $c \in \mathcal{C}$

$\alpha_{g_P}^f$ Number of aircraft of type $f \in F$ located at gateway $g \in G$ at the beginning of the Next-Day Air network

$\alpha_{g_D}^f$ Number of aircraft of type $f \in F$ located at gateway $g \in G$ at the end of the Next-Day Air network

Indicators:

β_i^r Indicates the origin and destination of route $r \in R$

γ_c^e Amount of extreme route $e \in E$ included in composite $c \in \mathcal{C}$

γ_c^f Number of aircraft of type $f \in F$ included in composite $c \in \mathcal{C}$

$\gamma_c^f(\bar{i})$ Number of aircraft of type $f \in F$ included in composite $c \in \mathcal{C}$ departing from location $i \in G \cup H$

$\gamma_c^f(\underline{i})$ Number of aircraft of type $f \in F$ included in composite $c \in \mathcal{C}$ terminating at location $i \in G \cup H$

γ_c^{fr} Number of aircraft route (f,r) included in composite $c \in \mathcal{C}$

δ_e^{fr} Indicates the aircraft route (f,r) corresponding to extreme route $e \in E$

δ_{ij}^{fr} Indicates the arcs $(i,j) \in A$ through which aircaft route (f,r) passes

δ_c^{gh} Indicates the gateway-hub commodities $(g,h) \in \mathcal{K}$ that are covered by composite $c \in \mathcal{C}$

δ_{ij}^h Indicates the arc $(i,j) \in A$ associated with hub sort

δ_{ij}^p Indicates the arcs $(i,j) \in A$ through which package flow path p passes

δ_{ij}^{qk} Indicates the arcs $(i,j) \in A$ contained in path $q^k \in q$

δ_h^r Indicates the hub sort arc through which route $r \in R$ passes

Decision Variables:

v_c Amount of composite $c \in C$ selected in **ARM**

w_e Amount of extreme route $e \in E$ selected in **RO**

x_{ij}^k Amount of commodity $k \in K$ assigned to arc $(i,j) \in A$

y_{ij} Binary decision about opening arc $(i,j) \in A$ in general network design problems

y_{ij}^f Number of units of facility type $f \in F$ assigned to arc $(i,j) \in A$ in network loading problems

y_r^f Number of aircraft route (f,r) selected

Dual Variables:

μ_{gh}^P Dual variables associated with (g,h) pickup capacity-demand constraint in **RO**

μ_{gh}^D Dual variables associated with (g,h) delivery capacity-demand constraint in **RO**

π_h^a **ESSND** dual variable associated with landing constraint for hub $h \in H$

π_{if}^b **ESSND** dual variable associated with aircraft balance constraint for location $i \in G \cup H$ and fleet type $f \in H$

π_f^n **ESSND** dual variable associated with plane count constraint for fleet type $f \in F$

p_h^a **ARM** dual variable associated with landing constraint for hub $h \in H$

p_{if}^b **ARM** dual variable associated with aircraft balance constraint for location $i \in G \cup H$ and fleet type $f \in F$

p_f^n **ARM** dual variable associated with plane count constraint for fleet type $f \in F$

q_k **ARM** dual variable associated with convexity constraint $k \in \mathcal{K}$

σ_k **ESSND** dual variable associated with convexity constraint $k \in \mathcal{K}$

Appendix B

Formulations

Express Shipment Service Network Design, General Form (**ESSND**)

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k + \sum_{f \in F} \sum_{r \in R^f} d_r^f y_r^f$$

$$\text{subject to: } \sum_{k \in K} x_{ij}^k \leq \sum_{f \in F} \sum_{r \in R^f} \delta_{ij}^{fr} u_r^f y_r^f \quad (i,j) \in A$$

$$\sum_{j:(i,j) \in A} x_{ij}^k - \sum_{j:(j,i) \in A} x_{ji}^k = \begin{cases} b^k & \text{if } i = O(k) \\ -b^k & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \quad i \in N,\ k \in K$$

$$\sum_{r \in R^f} \beta_i^r y_r^f = 0 \quad i \in N,\ f \in F$$

$$\sum_{k \in K} \sum_{(i,j) \in A} \delta_{ij}^h x_{ij}^k \leq e_h \quad h \in H$$

$$\sum_{r \in R^f} y_r^f \leq n_f \quad f \in F$$

$$\sum_{f \in F} \sum_{r \in R^f} \delta_h^r y_r^f \leq a_h \quad h \in H$$

$$x_{ij}^k \geq 0 \quad (i,j) \in A,\ k \in K$$

$$y_r^f \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F$$

Express Shipment Service Network Design
Path-Based Carrier-Specific Form (**ESSND-C**)

$$\min \sum_{f\in F}\sum_{r\in R^f} d_r^f y_r^f$$

subject to:

$$\begin{aligned}
\sum_{k\in K}\sum_{p\in P^k}\delta_{ij}^p b^k x_p^k &\leq \sum_{f\in F}\sum_{r\in R^f}\delta_{ij}^r u_r^f y_r^f && \text{for all } (i,j)\in A\\
\sum_{p\in P^k} x_p^k &= 1 && \text{for all } k\in K\\
\sum_{k\in K}\sum_{p\in P^k}\delta_h^p b^k x_p^k &\leq e_h && h\in H\\
\sum_{r\in R_P^f(\overline{g})} y_r^f - \sum_{r\in R_D^f(\underline{g})} y_r^f &= 0 && g\in G,\ f\in F\\
\sum_{r\in R_D^f(\overline{h})} y_r^f - \sum_{r\in R_P^f(\underline{h})} y_r^f &= 0 && h\in H,\ f\in F\\
\sum_{r\in R_P^f} y_r^f &\leq n_f && f\in F\\
\sum_{r\in R_D^f} y_r &\leq n_f && f\in F\\
\sum_{f\in F}\sum_{r\in R_P^f(\underline{h})} y_r^f &\leq a_h && h\in H\\
x_p^k &\geq 0 && p\in P^k,\ k\in K\\
y_r^f &\in \mathbb{Z}_+ && r\in R^f,\ f\in F
\end{aligned}$$

Express Shipment Service Network Design
No Ramp Transfer Form (**ESSND-R**)

$$\min \mathbf{d}'\mathbf{y}$$

$$\text{subject to} \quad \sum_{(g,h)\in\mathcal{K}} x_r^{gh} - \sum_{f\in F} u_r^f y_r^f \leq 0 \quad r \in R$$

$$\sum_{r\in R_P(g,h)} x_r^{gh} = b_P^{gh} \quad (g,h) \in \mathcal{K}_P$$

$$\sum_{r\in R_D(g,h)} x_r^{gh} = b_D^{gh} \quad (g,h) \in \mathcal{K}_D$$

$$\mathbf{B}\mathbf{y} = \mathbf{0}$$

$$\mathbf{A}\mathbf{y} \leq \mathbf{a}$$

$$\mathbf{N}\mathbf{y} \leq \mathbf{n}$$

$$y_r^f \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F$$

$$x_r^{gh} \geq 0 \quad r \in R^f,\ (g,h) \in \mathcal{K}$$

Routes-Only Model (**RO**)

$$\min \widehat{\mathbf{d}}'\mathbf{w}$$

$$
\begin{aligned}
\text{subject to} \quad & \sum_{e \in E_P} \widehat{u}^e_{gh} w_e \geq b^{gh}_P \quad (g,h) \in \mathcal{K}_P \\
& \sum_{e \in E_D} \widehat{u}^e_{gh} w_e \geq b^{gh}_D \quad (g,h) \in \mathcal{K}_D \\
& \widehat{\mathbf{B}}\mathbf{w} = \mathbf{0} \\
& \widehat{\mathbf{A}}\mathbf{w} \leq \mathbf{a} \\
& \widehat{\mathbf{N}}\mathbf{w} \leq \mathbf{n} \\
& \sum_{e \in E} \delta^{fr}_e w_e \in \mathbb{Z}_+ \quad r \in R^f,\ f \in F
\end{aligned}
$$

Aircraft Routing Model (**ARM**)
Set Covering Form

$$\min \overline{\mathbf{d}}'\mathbf{v}$$

subject to

$$\sum_{c \in \mathcal{C}_P} \delta_c^{gh} v_c \geq 1 \quad (g,h) \in \mathcal{K}_P$$

$$\sum_{c \in \mathcal{C}_D} \delta_c^{gh} v_c \geq 1 \quad (g,h) \in \mathcal{K}_D$$

$$\overline{\mathbf{B}}\mathbf{v} = \mathbf{0}$$

$$\overline{\mathbf{A}}\mathbf{v} \leq \mathbf{a}$$

$$\overline{\mathbf{N}}\mathbf{v} \leq \mathbf{n}$$

$$\sum_{c \in \mathcal{C}} \gamma_c^{fr} v_c \in \mathbb{Z}_+ \quad r \in R^f, \ f \in F$$

180

Bibliography

[1] A. Agrawal, P. Klein, and R. Ravi. "When Trees Collide: An Approximation Algorithm for the Generalized Steiner Problem on Networks.," *SIAM Journal on Computing*, *24*(3):440–456 (1995).

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.

[3] D. Alevras, M. Grötschel, and R. Wessaly. "Cost-Efficient Network Synthesis from Leased Lines," *Annals of Operations Research*, *76*:1–20 (1998).

[4] A. Atamtürk. *On Capacitated Network Design Cut-Set Polyhedra*. Technical Report, University of California at Berkeley, 1999.

[5] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. "A Dual-Based Algorithm for Multi-Level Network Design," *Management Science*, *40*(5):567–581 (1994).

[6] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. "Modeling and Heuristic Worst-Case Performance Analysis of The Two-Level Network Design Problem," *Management Science*, *40*(7):846–867 (1994).

[7] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. "Heuristics, LPs, and Trees on Trees: Network Design Analysis," *Operations Research*, *44*(3):478–496 (1996).

[8] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. "Designing Hierarchical Survivable Networks," *Operations Research*, *46*(1):116–136 (1998).

[9] A. Balakrishnan, T. L. Magnanti, and R. T. Wong. "A Decomposition Algorithm for Local Access Telecommunications Network Expansion Planning," *Operations Research*, *43*(1):58–76 (1995).

[10] F. Barahona. "Network Design Using Cut Inequalities," *SIAM Journal on Optimization*, *6*(3):823–837 (1996).

[11] C. Barnhart. "Dual-Ascent Methods for Large-Scale Multicommodity Flow Problems.," *Naval Research Logistics*, *40*(3):305–324 (1993).

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

[13] C. Barnhart, C. A. Hane, E. L. Johnson, and G. Sigismondi. "A Column Generation and Partitioning Approach for Multicommodity Flow Problems," *Telecommunication Systems*, *3*(3-4):239–258 (1995).

[14] C. Barnhart, C. A. Hane, and P. H. Vance. "Integer Multicommodity Flow Problems," *Lecture Notes in Economics and Mathematical Systems*, *450*:17–31 (1997).

[15] C. Barnhart, C. A. Hane, and P. H. Vance. "Using Branch-and-Price-and-Cut to Solve Origin-Destination Integer Multicommodity Flow Problems," *Operations Research*, *48*(2):318–326 (2000).

[16] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. F. Savelsbergh, and P. H. Vance. "Branch and Price: Column Generation for Solving Huge Integer Programs," *Operations Research*, *46*(3):316–329 (1998).

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

[18] C. Barnhart and Y. Sheffi. "A Network Based Primal-Dual Heuristic for the Solution of Multicommodity Network Flow Problems.," *Transportation Science*, *27*(2):102–117 (1993).

[19] C. Barnhart and R. G. Shenoi. "An Approximate Model and Solution Approach for the Long-Haul Crew Pairing Problem," *Transportation Science*, *32*(3):221–231 (1998).

[20] D. Bertsimas and C. P. Teo. "From Valid Inequalities to Heuristics: A Unified View of Primal-Dual Approximation Algorithms in Covering Problems," *Operations Research*, *46*(4):503–514 (1998).

[21] D. Bienstock, S. Chopra, O. Günlük, and C. Y. Tsai. "Minimum Cost Capacity Installation for Multicommodity Networks," *Mathematical Programming*, *81*(2):177–199 (1998).

[22] D. Bienstock and O. Günlük. "Capacitated Network Design - Polyhedral Structure and Computation," *INFORMS Journal on Computing*, *8*(3):243–259 (1996).

[23] D. Bienstock and O. Günlük. "Computational Experience with a Difficult Mixed-Integer Multicommodity Flow Problem," *Mathematical Programming*, *68*(2):213–237 (1995).

[24] S. Chopra, L. Gilboa, and S. T. Sastry. "Source Sink Flows with Capacity Installation in Batches," *Discrete Applied Mathematics*, *85*(3):165–192 (1998).

[25] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. New York: Wiley, 1998.

[26] T. G. Crainic. "Service Network Design in Freight Transportation," *European Journal of Operational Research*, *122*(2):272–288 (2000).

[27] H. Crowder, E. L. Johnson, and M. W. Padberg. "Soving Large-Scale Zero-One Linear Programming Problems," *Operations Research*, *31*(5):803–834 (1983).

[28] G. B. Dantzig and P. Wolfe. "Decomposition Principle for Linear Programs.," *Operations Research*, *8*:101–111 (1960).

[29] J. M. Farvolden and W. B. Powell. "Subgradient Methods for the Service Network Design Problem," *Transportation Science*, *28*(3):256–272 (1994).

[30] J. M. Farvolden, W. B. Powell, and I. J. Lustig. "A Primal Partitioning Solution for the Arc-Chain Formulation of a Multicommodity Network Flow Problem," *Operations Research*, *41*(4):669–693 (1993).

[31] H. N. Gabow, M. X. Goemans, and D. P. Williamson. "An Efficient Approximation Algorithm for the Survivable Network Design Problem," *Mathematical Programming*, *82*(1-2):13–40 (1998).

[32] V. Gabrel, A. Knippel, and M. Minoux. "Exact Solution of Multicommodity Network Optimization Problems with General Step Cost Functions," *Operations Research Letters*, *25*(1):15–23 (1999).

[33] M. X. Goemans and D. J. Bertsimas. "Survivable Networks, Linear Programming Relaxations and the Parsimonious Property," *Mathematical Programming*, *60*(2):145–166 (1993).

[34] M. X. Goemans and D. P. Williamson. "A General Approximation Technique for Constrained Forest Problems," *SIAM Journal on Computing*, *24*(2):296–317 (1995).

[35] R. Gopalan and K. Talluri. "The Aircraft Maintenance Routing Problem," *Operations Research*, *46*(2):260–271 (1998).

[36] M. F. Gorman. "An Application of Genetic and Tabu Searches to the Freight Railroad Operating Plan Problem," *Annals of Operations Research*, *78*:51–69 (1998).

[37] M. F. Gorman. "Santa Fe Railway Uses an Operating-Plan Model to Improve its Service Design," *Interfaces*, *28*(4):1–12 (1998).

[38] M. Göthe-Lundgren and T. Larsson. "A Set Covering Reformulation of The Pure Fixed Charge Transportation Problem," *Discrete Applied Mathematics*, *48*(3):245–259 (1994).

[39] T. Grünert and H. J. Sebastian. "Planning Models for Long-Haul Operations of Postal and Express Shipment Companies," *European Journal of Operatoinal Research*, *122*(2):289–309 (2000).

[40] M. Guignard and S. Kim. "Lagrangean Decomposition - A Model Yielding Stronger Lagrangean Bounds," *Mathematical Programming*, *39*(2):215–228 (1987).

[41] O. Günlük. "A Branch-and-Cut Algorithm for Capacitated Network Design Problems," *Mathematical Programming*, *86*(1):17–39 (1999).

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

[43] D. S. Hochbaum and J. S. Naor. "Approximation Algorithms for Network Design Problems on Bounded Subsets," *Journal of Algorithms*, *21*(2):403–414 (1996).

[44] K. L. Hoffman and M. Padberg. "Solving Airline Crew Scheduling Problems by Branch-and-Cut," *Management Science*, *39*(6):657–682 (1993).

[45] J. Holmberg and J. Hellstrand. "Solving the Uncapacitated Network Design Problem by a Lagrangian Heuristic and Branch-and-Bound," *Operations Research*, *46*(2):247–259 (1998).

[46] ILOG. *CPLEX 6.5 User's Manual*, 1999.

[47] K. Jain. "A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem." *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS '98)*. 1998.

[48] K. L. Jones, I. J. Lustig, J. M. Farvolden, and W. B. Powell. "Multicommodity Network Flows: The Impact of Formulation on Decomposition," *Mathematical Programming Study*, *62*(1):95–117 (1993).

[49] D. R. Karger. "Random Sampling in Cut, Flow, and Network Design Problems," *Mathematics of Operations Research*, *24*(2):383–413 (1999).

[50] D. Kim. *Large Scale Transportation Service Network Design: Models, Algorithms and Applications*. PhD dissertation, Massachusetts Institute of Technology, 1997.

[51] D. Kim and C. Barnhart. "Transportation Service Network Design: Models and Algorithms," *Lecture Notes in Economics and Mathematical Systems, Computer-Aided Transit Scheduling*, *471*:259–283 (1999).

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

[53] N. Krishnan, C. Barnhart, D. Kim, and K. Ware. "Network Design for Express Shipment Delivery," *Computational Optimization and Applications* (to appear).

[54] P. J. Lederer and R. S. Nambimadom. "Airline Network Design," *Operations Research*, *46*(6):785–804 (1998).

[55] T. Leighton, F. Makedon, S. Plotkin, C. Stein, E. Tardos, and S. Tragoudas. "Fast Approximation Algorithms for Multicommodity Flow Problems.," *Journal of Computer and System Sciences*, *50*(2):228–243 (1995).

[56] T. L. Magnanti and P. Mirchandani. "Shortest Paths, Single Origin-Destination Network Design and Associated Polyhedra," *Networks*, *23*(2):103–121 (1993).

[57] T. L. Magnanti, P. Mirchandani, and R. Vachani. "The Convex Hull of Two Core Capacitated Network Design Problems," *Mathematical Programming*, *26*(2):233–250 (1993).

[58] T. L. Magnanti, P. Mirchandani, and R. Vachani. "Modeling and Solving the Two-Facility Capacitated Network Loading Problem," *Operations Research*, *43*(1):142–157 (1995).

[59] T. L. Magnanti, P. Mireault, and R. T. Wong. "Tailoring Benders Decomposition for Uncapacitated Network Design Problems.," *Mathematical Programming Study*, *26*:112–154 (1986).

[60] T. L. Magnanti and R. T. Wong. "Network Design and Transporation Planning: Models and Algorithms," *Transporation Science*, *18*(1):1–55 (1984).

[61] M. Minoux. "Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications," *Networks*, *19*(3):313–360 (1989).

[62] Y. S. Myung, H. J. Kim, and D. W. Tcha. "Design of Communication Network with Survivability Constraints," *Management Science*, *45*(2):238–252 (1999).

[63] G. L. Nemhauser, M. W. P. Savelsbergh, and G. S. Sigismondi. "MINTO, a Mixed INTeger Optimizer," *Operations Research Letters*, *15*(1):47–58 (1994).

[64] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley and Sons, Inc., 1988.

[65] H. N. Newton, C. Barnhart, and P. H. Vance. "Constructing Railroad Blocking Plans to Minimize Handling Costs," *Transportation Science*, *32*(4):330–45 (1998).

[66] M. W. Padberg, T. J. VanRoy, and L. A. Wolsey. "Valid Linear Inequalities for Fixed Charge Problems," *Operations Research*, *33*(4):842–861 (1985).

[67] Y. Pochet and L. A. Wolsey. "Integer Knapsack and Flow Covers with Divisible Coefficients: Polyhedra, Optimization, and Separation," *Discrete Applied Mathematics*, *59*(1):57–74 (1995).

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

[69] J. Stallaert. "Valid Inequalities and Separation for Capacitated Fixed Charge Flow Problems," *Discrete Applied Mathematics*, *98*(3):265–274 (2000).

[70] T. J. VanRoy and L. A. Wolsey. "Valid Inequalities and Separation for Uncapacitated Fixed Charge Networks," *Operations Research Letters*, *4*(3):105–112 (1985).

[71] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. "A Primal-Dual Approximation Algorithm for Generalized Steiner Network Problems," *Combinatorica*, *15*(3):435–454 (1995).

[72] L. A. Wolsey. "Faces of Linear Inequalities in 0-1 Variables," *Mathematical Programming*, *8*:165–178 (1975).

[73] K. Ziarati, F. Soumis, J. Desrosiers, S. Gelinas, and A. Saintonge. "Locomotive Assignment with Heterogeneous Consists at CN North America," *European Journal of Operational Research*, *97*(2):281–292 (1997).

[74] K. Ziarati, F. Soumis, J. Desrosiers, and M. M. Solomon. "A Branch-First, Cut-Second Approach for Locomotive Assignment," *Management Science*, *45*(8):1156–1168 (1999).